



ISO/IEC 24775-6

Edition 1.0 2014-11

INTERNATIONAL STANDARD



Information technology – Storage management –
Part 6: Fabric

IECNORM.COM : Click to view the full PDF of ISO/IEC 24775-6:2014



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 ISO/IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about ISO/IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

IECNORM.COM : Click to view the full text of ISO/IEC 24775-6:2014



ISO/IEC 24775-6

Edition 1.0 2014-11

INTERNATIONAL STANDARD



Information technology – Storage management –
Part 6: Fabric

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE

H

ICS 35.200

ISBN 978-2-8322-1954-6

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD.....	21
Introduction	23
1 Scope	25
2 Normative references	25
3 Terms and definitions	25
4 Typographical conventions.....	26
4.1 Maturity model.....	26
4.2 Experimental maturity level	26
4.3 Implemented maturity level	26
4.4 Stable maturity level.....	27
4.5 Finalized maturity level.....	27
4.6 Deprecated Material.....	27
5 Fabric Profile	29
5.1 Description	29
5.1.1 General.....	29
5.1.2 SANS and Fabrics as AdminDomains.....	29
5.1.3 Fabrics and Topology.....	30
5.1.4 Systems and NetworkPorts	30
5.1.5 N Port Virtualization (NPIV).....	31
5.1.6 Zoning	32
5.1.7 Conditional Classes and Properties	33
5.1.8 Fabric Predefined Indications.....	34
5.2 Health and Fault Management.....	35
5.3 Cascading Considerations	35
5.4 Supported Subprofiles and Package.....	35
5.5 Methods of this Profile.....	35
5.6 Client Considerations and Recipes	36
5.6.1 Fabric Identifier.....	36
5.6.2 FCPort OperationalStatus	36
5.6.3 Switch ComputerSystem OperationalStatus	36
5.6.4 Discover The Fabric Topology	37
5.7 Registered Name and Version.....	41
5.8 CIM Elements.....	41
5.8.1 General.....	41
5.8.2 CIM_ActiveConnection.....	47
5.8.3 CIM_AdminDomain (Fabric).....	47
5.8.4 CIM_AdminDomain (SAN)and	48
5.8.5 CIM_Component (Platform to Fabric)	49
5.8.6 CIM_Component (Switch to Fabric)	49
5.8.7 CIM_ComputerSystem (Host Platform).....	49
5.8.8 CIM_ComputerSystem (Partitioned Switch).....	50
5.8.9 CIM_ComputerSystem (Storage Platform).....	51
5.8.10 CIM_ComputerSystem (Switch).....	51
5.8.11 CIM_ConnectivityCollection	52
5.8.12 CIM_ContainedDomain	53
5.8.13 CIM_DeviceSAPImplementation (Non-Switch to FCPort).....	53
5.8.14 CIM_DeviceSAPImplementation (Switch to FCPort).....	53
5.8.15 CIM_ElementCapabilities (ZoneCapabilities to Fabric).....	54
5.8.16 CIM_ElementCapabilities (ZoneCapabilities to Switch)	54
5.8.17 CIM_ElementSettingData (ZoneMembershipSettingData to Zone).....	55
5.8.18 CIM_FCActiveConnection	55
5.8.19 CIM_FCPort (Host FCPort)	55

5.8.20	CIM_FCPort (Host NPIV FCPort).....	56
5.8.21	CIM_FCPort (Partitioned Switch FCPort).....	57
5.8.22	CIM_FCPort (Storage FCPort).....	59
5.8.23	CIM_FCPort (Switch FCPort).....	59
5.8.24	CIM_FilterCollection (Fabric Predefined FilterCollection).....	61
5.8.25	CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint).....	61
5.8.26	CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint).....	61
5.8.27	CIM_HostedCollection (Fabric to ConnectivityCollection).....	62
5.8.28	CIM_HostedCollection (Fabric to predefined FilterCollection).....	62
5.8.29	CIM_HostedCollection (System to LogicalPortGroup).....	62
5.8.30	CIM_HostedCollection (Zones or ZoneSets to Fabric).....	63
5.8.31	CIM_HostedCollection (Zones or ZoneSets to Switch).....	63
5.8.32	CIM_HostedDependency.....	64
5.8.33	CIM_IndicationFilter (Fabric ComputerSystem Creation).....	64
5.8.34	CIM_IndicationFilter (Fabric ComputerSystem Deletion).....	65
5.8.35	CIM_IndicationFilter (Fabric ComputerSystem OperationalStatus).....	66
5.8.36	CIM_IndicationFilter (Fabric FCPort Creation).....	67
5.8.37	CIM_IndicationFilter (Fabric FCPort Deletion).....	68
5.8.38	CIM_IndicationFilter (Fabric FCPort OperationalStatus).....	69
5.8.39	CIM_IndicationFilter (Fabric Fabric Added or Removed).....	70
5.8.40	CIM_IndicationFilter (Fabric Fabric Merge or Segmentation).....	71
5.8.41	CIM_IndicationFilter (Fabric Switch Added or Removed).....	72
5.8.42	CIM_IndicationFilter (Fabric Switch Status Changed).....	72
5.8.43	CIM_IndicationFilter (Fabric Zone Database Changed).....	73
5.8.44	CIM_IndicationFilter (Fabric ZoneSet Activated).....	74
5.8.45	CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus).....	75
5.8.46	CIM_IndicationFilter (WQL Fabric FCPort OperationalStatus).....	76
5.8.47	CIM_LogicalPortGroup.....	77
5.8.48	CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint).....	78
5.8.49	CIM_MemberOfCollection (LogicalPortGroup to FCPort).....	78
5.8.50	CIM_MemberOfCollection (Predefined Filter Collection to Fabric Filters).....	78
5.8.51	CIM_MemberOfCollection (ZoneSet to Zone).....	79
5.8.52	CIM_ProtocolEndpoint.....	79
5.8.53	CIM_SystemDevice (Non-Switch FCPort to Fabric).....	80
5.8.54	CIM_SystemDevice (Non-Switch FCPort to Platform).....	80
5.8.55	CIM_SystemDevice (Switch FCPort to Switch).....	80
5.8.56	CIM_Zone (Active).....	81
5.8.57	CIM_Zone (Inactive).....	81
5.8.58	CIM_ZoneCapabilities.....	81
5.8.59	CIM_ZoneMembershipSettingData.....	83
5.8.60	CIM_ZoneSet (Active).....	83
5.8.61	CIM_ZoneSet (Inactive).....	84
6	Enhanced Zoning and Enhanced Zone Control Subprofile.....	85
6.1	Description.....	85
6.2	Health and Fault Management.....	85
6.3	Cascading Considerations.....	85
6.4	Dependencies on Profiles, Subprofiles, and Packages.....	85
6.5	Methods of this Profile.....	85
6.5.1	CreateZoneAlias.....	85
6.5.2	AddZoneAlias.....	85
6.6	Client Considerations and Recipes.....	85
6.6.1	Create a ZoneAlias.....	85
6.6.2	Delete a ZoneAlias.....	87
6.7	Registered Name and Version.....	88

6.8	CIM Elements.....	89
6.8.1	General.....	89
6.8.2	CIM_ElementSettingData (ZoneMembershipSettingData to NamedAddressCollection)	89
6.8.3	CIM_HostedCollection (AdminDomain to Collection)	89
6.8.4	CIM_HostedCollection (ComputerSystem to Collection).....	90
6.8.5	CIM_MemberOfCollection	90
6.8.6	CIM_NamedAddressCollection	90
6.8.7	CIM_ZoneService (Zone Service)	91
7	Zone Control Subprofile.....	92
7.1	Description	92
7.2	Durable Names and Correlatable IDs of the Profile	92
7.3	Instrumentation Requirements	92
7.4	Health and Fault Management.....	92
7.5	Cascading Considerations	92
7.6	Dependencies on Profiles, Subprofiles, and Packages	92
7.7	Methods of this Profile.....	92
7.7.1	CreateZoneSet	92
7.7.2	CreateZone	93
7.7.3	CreateZoneMembershipSettingData	93
7.7.4	AddZone	93
7.7.5	AddZoneMembershipSettingData	93
7.7.6	ActivateZoneSetWithJob	93
7.7.7	ActivateZoneSet	94
7.7.8	SessionControlWithJob	94
7.7.9	SessionControl	95
7.7.10	Intrinsics for removing a zone from a zone set.....	96
7.7.11	Intrinsics for removing a zone alias from a zone	96
7.7.12	Intrinsics for removing a zone member from a zone or zone alias	96
7.7.13	Intrinsic for deleting a zone member	97
7.7.14	Intrinsic for deleting a zone, zone alias, or zone set	97
7.8	Client Considerations and Recipes	97
7.8.1	General.....	97
7.8.2	Create or delete zones Common Functions.....	97
7.8.3	Add new Zone Member to Existing Zone	100
7.8.4	Create new Zone, add new Zone Member, and add to existing ZoneSet	101
7.8.5	Create new ZoneSet and add existing Zone	103
7.8.6	Delete zone	105
7.8.7	Delete ZoneSet	106
7.8.8	Delete ZoneMember.....	107
7.9	Registered Name and Version	108
7.10	CIM Elements.....	108
7.10.1	General.....	108
7.10.2	CIM_HostedService (Fabric (AdminDomain) to ZoneService).....	108
7.10.3	CIM_HostedService (Switch (ComputerSystem) to ZoneService)	109
7.10.4	CIM_ZoneService (Zone Service)	109
8	FDMI Subprofile.....	111
8.1	Description	111
8.2	Health and Fault Management.....	111
8.3	Cascading Considerations	111
8.4	Dependencies on Profiles, Subprofiles, and Packages	111
8.5	Methods of this Profile.....	112
8.6	Client Considerations and Recipes	112
8.7	Registered Name and Version	112

8.8	CIM Elements.....	112
8.8.1	General.....	112
8.8.2	CIM_Component (Host Server to Fabric).....	112
8.8.3	CIM_ComputerSystem (Host Server).....	113
8.8.4	CIM_ControlledBy	113
8.8.5	CIM_ElementSoftwareIdentity.....	114
8.8.6	CIM_FCPort (Host FCPort)	114
8.8.7	CIM_HostedCollection (System to LogicalPortGroup)	115
8.8.8	CIM_InstalledSoftwareIdentity.....	115
8.8.9	CIM_MemberOfCollection (LogicalPortGroup to FCPort)	115
8.8.10	CIM_PhysicalPackage (HBA Package).....	116
8.8.11	CIM_PortController.....	116
8.8.12	CIM_Product (HBA Product)	117
8.8.13	CIM_ProductPhysicalComponent	117
8.8.14	CIM_Realizes	118
8.8.15	CIM_SoftwareIdentity (Driver).....	118
8.8.16	CIM_SoftwareIdentity (Firmware).....	118
8.8.17	CIM_SoftwareIdentity (Option ROM).....	119
8.8.18	CIM_SystemDevice (ComputerSystem to FCPort)	119
8.8.19	CIM_SystemDevice (ComputerSystem to PortController).....	120
9	Fabric Path Performance Subprofile	121
9.1	Description	121
9.2	Health and Fault Management.....	121
9.3	Dependencies on Profiles, Subprofiles, and Packages	121
9.4	Methods of this Profile.....	121
9.5	Client Considerations and Recipes	122
9.6	Registered Name and Version	122
9.7	CIM Elements.....	122
9.7.1	General.....	122
9.7.2	CIM_ElementStatisticalData.....	122
9.7.3	CIM_EndpointOfNetworkPipe	123
9.7.4	CIM_HostedCollection.....	123
9.7.5	CIM_HostedNetworkPipe	123
9.7.6	CIM_MemberOfCollection	124
9.7.7	CIM_Network.....	124
9.7.8	CIM_NetworkPipe	125
9.7.9	CIM_NetworkPortStatistics.....	125
9.7.10	CIM_ProtocolEndpoint	125
9.7.11	CIM_StatisticsCollection.....	126
10	Fibre Channel Security Subprofile.....	127
10.1	Description	127
10.2	Health and Fault Management Consideration.....	129
10.3	Cascading Considerations	129
10.4	Supported Profiles, Subprofiles, and Packages.....	129
10.5	Methods of the Profile	129
10.6	Client Considerations and Recipes	129
10.7	Registered Name and Version	129
10.8	CIM Elements.....	130
10.8.1	General.....	130
10.8.2	CIM_AuthorizationService	130
10.8.3	CIM_AuthorizedPrivilege	130
10.8.4	CIM_AuthorizedSubject.....	131
10.8.5	CIM_AuthorizedTarget	131
10.8.6	CIM_HostedService	131

10.8.7	CIM_ServiceAffectsElement (ManagedElement to Service)	132
10.8.8	CIM_ServiceAffectsElement (StorageHardwareID to Service)	132
10.8.9	CIM_ServiceAvailableToElement (Fabric AdminDomain to Service)	132
10.8.10	CIM_StorageHardwareID	133
11	Fabric Views Subprofile	134
11.1	Description	134
11.1.1	Synopsis	134
11.1.2	Overview	134
11.1.3	Topology View	135
11.1.4	FCSwitch View	136
11.2	Health and Fault Management Consideration	136
11.3	Cascading Considerations	136
11.4	Supported Profiles, Subprofiles, and Packages	136
11.5	Methods of the Profile	137
11.6	Client Considerations and Recipes	137
11.7	Registered Name and Version	137
11.8	CIM Elements	137
11.8.1	General	137
11.8.2	CIM_ElementCapabilities (View Capabilities)	138
11.8.3	SNIA_BaseInstance (View to FC Port)	138
11.8.4	SNIA_BaseInstance (View to Switch)	139
11.8.5	SNIA_ComponentView (FCSwitchView to Fabric)	139
11.8.6	SNIA_ComponentView (TopologyView to Fabric)	139
11.8.7	SNIA_FCSwitchView	140
11.8.8	SNIA_TopologyView	141
11.8.9	SNIA_ViewCapabilities	142
12	Virtual Fabrics Subprofile	143
12.1	Description	143
12.2	Health and Fault Management Consideration	145
12.3	Cascading Considerations	145
12.4	Supported Profiles, Subprofiles, and Packages	145
12.5	Methods of the Profile	145
12.6	Client Considerations and Recipes	145
12.7	Registered Name and Version	145
12.8	CIM Elements	145
12.8.1	General	145
12.8.2	CIM_Component (AdminDomain to Partitioning CS)	145
12.8.3	CIM_ElementConformsToProfile (SAN AdminDomain to Virtual Fabrics Registered-Profile)	146
13	Switch Profile	147
13.1	Description	147
13.1.1	General	147
13.1.2	FC Port Settings and Capabilities	148
13.1.3	Trunking	149
13.1.4	DetailedPortState, PortAvailability, OperationalStatus, and EnabledState	149
13.1.5	Conditional Classes and Properties	150
13.1.6	Switch Predefined Indications	151
13.2	Health and Fault Management	152
13.3	Cascading Considerations	153
13.4	Dependencies on Profiles, Subprofiles, and Packages	153
13.5	Methods of this Profile	153
13.6	Client Considerations and Recipes	153
13.6.1	Enable FCPort	153
13.6.2	Disable Port	154

13.6.3	Enable Switch	156
13.6.4	Disable Switch	157
13.6.5	Reset Switch	158
13.6.6	Set Port Speed	160
13.6.7	Set Port Type	161
13.6.8	Set Fibre Channel Switch Principal Priority	162
13.6.9	Set Switch Name	164
13.6.10	Set Port Name	165
13.6.11	Set Fibre Channel Switch Preferred Domain ID	166
13.6.12	Lock Fibre Channel Switch Domain ID	168
13.7	Registered Name and Version	170
13.8	CIM Elements	170
13.8.1	General	170
13.8.2	CIM_ComputerSystem (Partitioned Switch)	172
13.8.3	CIM_ComputerSystem (Switch)	173
13.8.4	CIM_ComputerSystemPackage	174
13.8.5	CIM_ElementCapabilities (FCPort to FCPortCapabilities)	175
13.8.6	CIM_ElementCapabilities (System to FCSwitchCapabilities)	175
13.8.7	CIM_ElementSettingData (FCPortSettings to FCPort)	175
13.8.8	CIM_ElementSettingData (FCSwitchSettings to ComputerSystem)	176
13.8.9	CIM_ElementStatisticalData (FCPortRateStatistics to FCPort)	176
13.8.10	CIM_ElementStatisticalData (FCPortStatistics to FCPort)	176
13.8.11	CIM_FCPort (Partitioned Switch FCPort)	177
13.8.12	CIM_FCPort (Switch FCPort)	178
13.8.13	CIM_FCPortCapabilities	179
13.8.14	CIM_FCPortRateStatistics	180
13.8.15	CIM_FCPortSettings	181
13.8.16	CIM_FCPortStatistics	181
13.8.17	CIM_FCSwitchCapabilities	183
13.8.18	CIM_FCSwitchSettings	184
13.8.19	CIM_FilterCollection (Switch Predefined FilterCollection)	184
13.8.20	CIM_HostedCollection (Redundancy Set)	185
13.8.21	CIM_HostedCollection (Statistics Collection)	185
13.8.22	CIM_HostedCollection (Switch to predefined FilterCollection)	185
13.8.23	CIM_IndicationFilter (Switch ComputerSystem Creation)	186
13.8.24	CIM_IndicationFilter (Switch ComputerSystem Deletion)	187
13.8.25	CIM_IndicationFilter (Switch ComputerSystem OperationalStatus)	187
13.8.26	CIM_IndicationFilter (Switch FCPort OperationalStatus)	188
13.8.27	CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus)	189
13.8.28	CIM_IndicationFilter (WQL Switch FCPort OperationalStatus)	190
13.8.29	CIM_MemberOfCollection (FCPort to RedundancySet)	191
13.8.30	CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection)	191
13.8.31	CIM_MemberOfCollection (Predefined Filter Collection to Switch Filters)	192
13.8.32	CIM_ProtocolEndpoint	192
13.8.33	CIM_RedundancySet	193
13.8.34	CIM_StatisticsCollection	193
13.8.35	CIM_SystemDevice	193
14	Switch Configuration Data Subprofile	195
14.1	Description	195
14.2	Durable Names and Correlatable IDs of the Profile	195
14.3	Instrumentation Requirements	195
14.4	Health and Fault Management	195
14.5	Cascading Considerations	195
14.6	Methods of this Profile – ApplyConfiguration	195

14.7	Client Considerations and Recipes	195
14.7.1	Get Switch Configuration	195
14.7.2	Set Switch Configuration	196
14.8	Registered Name and Version	196
14.9	CIM Elements	196
14.9.1	General	196
14.9.2	CIM_ComputerSystem	197
14.9.3	CIM_ConfigurationData	197
14.9.4	CIM_ElementSettingData	197
15	Blades Subprofile	199
15.1	Description	199
15.1.1	General	199
15.1.2	Instance Diagram	199
15.1.3	Blades Predefined Indications	199
15.2	Health and Fault Management	200
15.3	Cascading Considerations	201
15.4	Dependencies on Profiles, Subprofiles, and Packages	201
15.5	Methods of this Profile	201
15.6	Client Considerations and Recipes	201
15.7	Registered Name and Version	201
15.8	CIM Elements	201
15.8.1	General	201
15.8.2	CIM_FilterCollection (Blades Predefined FilterCollection)	202
15.8.3	CIM_HostedCollection (Switch to Blades predefined FilterCollection)	203
15.8.4	CIM_IndicationFilter (Blades LogicalModule Creation)	203
15.8.5	CIM_IndicationFilter (Blades LogicalModule Deletion)	204
15.8.6	CIM_IndicationFilter (Blades LogicalModule OperationalStatus)	205
15.8.7	CIM_IndicationFilter (WQL Blades LogicalModule OperationalStatus)	206
15.8.8	CIM_LogicalModule	207
15.8.9	CIM_MemberOfCollection (Blades FilterCollection to Switch FilterCollection)	207
15.8.10	CIM_MemberOfCollection (Predefined Filter Collection to Blades Filters)	208
15.8.11	CIM_ModulePort	208
15.8.12	CIM_PhysicalPackage (Logical Module)	208
15.8.13	CIM_Product (Blade)	209
15.8.14	CIM_ProductPhysicalComponent	209
15.8.15	CIM_Realizes (Logical Module Package)	209
15.8.16	CIM_SystemDevice (Logical Module)	210
16	Switch Partitioning Subprofile	211
16.1	Description	211
16.2	Health and Fault Management Consideration	212
16.3	Cascading Considerations	212
16.4	Supported Profiles, Subprofiles, and Packages	212
16.5	Methods of the Profile	213
16.6	Client Considerations and Recipes	213
16.7	Registered Name and Version	213
16.8	CIM Elements	213
16.8.1	General	213
16.8.2	CIM_ComputerSystem (Partitioned)	213
16.8.3	CIM_ComputerSystem (Partitioning)	214
16.8.4	CIM_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities)	214
16.8.5	CIM_ElementConformsToProfile (Partitioning ComputerSystem to Switch Partitioning RegisteredProfile)	215
16.8.6	CIM_ElementSettingData	

(Association between NetworkPort and NetworkPortSettings)	215
16.8.7 CIM_FCPort (Partitioned)	216
16.8.8 CIM_HostedDependency (NetworkPort to FCPort)	216
16.8.9 CIM_HostedDependency (Partitioning CS to Partitioned CS)	217
16.8.10 CIM_NetworkPort (Partitioning)	217
16.8.11 CIM_NetworkPortCapabilities	217
16.8.12 CIM_NetworkPortSettings	218
16.8.13 CIM_SystemDevice (FCPort to Partitioned ComputerSystem)	218
16.8.14 CIM_SystemDevice (NetworkPort to ComputerSystem)	219
17 Extender Profile	220
17.1 Description	220
17.1.1 General	220
17.1.2 FC Extender Node Topology Classes	220
17.1.3 FC Extender Node Network Connectivity Classes	221
17.1.4 FC Extender Group Network Connectivity Classes	221
17.1.5 Extender Predefined Indications	222
17.2 Health and Fault Management	223
17.3 Cascading Considerations	223
17.4 Supported Subprofiles and Packages	224
17.5 Methods of this Profile	224
17.6 Client Considerations and Recipes	224
17.6.1 Extender Connectivity Settings	224
17.6.2 Extender Connective Statistics	226
17.6.3 Extender Port Group Information	228
17.6.4 Extender Topology Mapping	230
17.7 Registered Name and Version	234
17.8 CIM Elements	234
17.8.1 General	234
17.8.2 CIM_BindsTo (IPPE to PE)	237
17.8.3 CIM_BindsTo (IPPE to RemoteSAP)	237
17.8.4 CIM_BindsTo (IPPE to RemoteSAP)	238
17.8.5 CIM_BindsTo (PE to RemotePort)	238
17.8.6 CIM_BindsTo (TCPPE to IPPE)	238
17.8.7 CIM_BindsTo (TCPPE to PE)	239
17.8.8 CIM_BindsTo (TCPPE to RemotePort)	239
17.8.9 CIM_Component	239
17.8.10 CIM_ComputerSystem (Extender Node)	240
17.8.11 CIM_DeviceSAPImplementation	240
17.8.12 CIM_ElementSettingData (IPPE to IPSettings)	241
17.8.13 CIM_ElementSettingData (PE to FCIPSettings)	241
17.8.14 CIM_ElementSettingData (System to FCIPSettings)	241
17.8.15 CIM_ElementSettingData (System to IPSettings)	242
17.8.16 CIM_ElementSettingData (System to TCPSettings)	242
17.8.17 CIM_ElementSettingData (TCPPE to TCPSettings)	242
17.8.18 CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics)	243
17.8.19 CIM_ElementStatisticalData (FCPort to FCPortStatistics)	243
17.8.20 CIM_ElementStatisticalData (IPPE to IPEndpointStatistics)	243
17.8.21 CIM_ElementStatisticalData (System to TCPStatisticalData)	244
17.8.22 CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics)	244
17.8.23 CIM_EndpointOfNetworkPipe (PE to NetworkPipe)	244
17.8.24 CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe)	245
17.8.25 CIM_EthernetPort	245
17.8.26 CIM_EthernetPortStatistics	246
17.8.27 CIM_FCIPSettings	246

17.8.28	CIM_FCPort	247
17.8.29	CIM_FCPortStatistics	248
17.8.30	CIM_FilterCollection (Extender Predefined FilterCollection)	250
17.8.31	CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)	250
17.8.32	CIM_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint)	250
17.8.33	CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)	251
17.8.34	CIM_HostedCollection (Extender to predefined FilterCollection)	251
17.8.35	CIM_HostedNetworkPipe	251
17.8.36	CIM_IPEndpointStatistics	252
17.8.37	CIM_IPProtocolEndpoint	252
17.8.38	CIM_IPSettings	253
17.8.39	CIM_IndicationFilter (Extender ComputerSystem Creation)	253
17.8.40	CIM_IndicationFilter (Extender ComputerSystem Deletion)	254
17.8.41	CIM_IndicationFilter (Extender ComputerSystem OperationalStatus)	255
17.8.42	CIM_IndicationFilter (WQL Extender ComputerSystem OperationalStatus)	256
17.8.43	CIM_LANEndpoint	257
17.8.44	CIM_MemberOfCollection (Predefined Filter Collection to Extender Filters)	258
17.8.45	CIM_Network	258
17.8.46	CIM_NetworkPipe	258
17.8.47	CIM_NetworkPipeComposition	259
17.8.48	CIM_ProtocolEndpoint	259
17.8.49	CIM_RemotePort	259
17.8.50	CIM_RemoteServiceAccessPoint	260
17.8.51	CIM_SystemDevice (System to EthernetPort)	260
17.8.52	CIM_SystemDevice (System to FCPort)	261
17.8.53	CIM_TCPEndpointStatistics	261
17.8.54	CIM_TCPProtocolEndpoint	261
17.8.55	CIM_TCPSettings	262
17.8.56	CIM_TCPStatisticalData	262
18	Router Profile	264
19	SAS Expander Profile	265
19.1	Description	265
19.2	Health and Fault Management Consideration	266
19.3	Cascading Considerations	266
19.4	Supported Profiles, Subprofiles, and Packages	266
19.4.1	General	266
19.4.2	Enclosure	266
19.4.3	Software Inventory and Software Update	266
19.5	Methods of the Profile	266
19.6	Client Considerations and Recipes	266
19.7	CIM Elements	266
19.8	Registered Name and Version	266
19.9	CIM Elements	267
19.9.1	General	267
19.9.2	CIM_ComputerSystem	267
19.9.3	CIM_ConcreteComponent	268
19.9.4	CIM_ElementSoftwareIdentity	268
19.9.5	CIM_Product	268
19.9.6	CIM_ProductElementComponent	269
19.9.7	CIM_SASPort	269
19.9.8	CIM_SoftwareIdentity	269
19.9.9	CIM_SystemDevice (Port)	270
19.9.10	CIM_SystemDevice (SASPHY)	270
19.9.11	SNIA_SASPHY	270

20	N Port Virtualizer Profile	272
20.1	Description	272
20.2	Implementation.....	272
20.3	Health and Fault Management Consideration.....	273
20.4	Cascading Considerations	273
20.5	Supported Profiles, Subprofiles, and Packages.....	273
20.6	Methods of the Profile	273
20.7	Client Considerations and Recipes	273
20.8	Registered Name and Version	273
20.9	CIM Elements.....	273
20.9.1	General.....	273
20.9.2	CIM_Component (N Port Virtualizer to Fabric).....	274
20.9.3	CIM_ComputerSystem (N Port Virtualizer).....	274
20.9.4	CIM_ComputerSystemPackage (N Port Virtualizer to Physical Package).....	275
20.9.5	CIM_DeviceSAPImplementation (ProtocolEndpoint to Gateway FCPort).....	275
20.9.6	CIM_DeviceSAPImplementation (ProtocolEndpoint to NPIV FCPort).....	276
20.9.7	CIM_FCActiveConnection (Gateway)	276
20.9.8	CIM_FCActiveConnection (N Port Virtualization).....	277
20.9.9	CIM_FCPort (Fabric NPIV).....	277
20.9.10	CIM_FCPort (Gateway).....	278
20.9.11	CIM_HostedAccessPoint (N Port Virtualizer System to ProtocolEndpoint)	279
20.9.12	CIM_LogicalIdentity (NPIV Port to Switch Port).....	279
20.9.13	CIM_ProtocolEndpoint (N Port Virtualizer).....	280
20.9.14	CIM_SystemDevice (N Port Virtualizer Gateway FCPort to Gateway System)	280
20.9.15	CIM_SystemDevice (N Port Virtualizer NPIV FCPort to Gateway System)	281
21	Inter Fabric Routing Profile.....	282
21.1	Description	282
21.1.1	General.....	282
21.1.2	Switch Topology	282
21.1.3	Identification of Ports	283
21.2	Health and Fault Management Consideration.....	284
21.3	Cascading Considerations	284
21.4	Supported Profiles, Subprofiles, and Packages.....	284
21.5	Methods of the Profile	284
21.6	Client Considerations and Recipes	284
21.7	Registered Name and Version	284
21.8	CIM Elements.....	285
21.8.1	General.....	285
21.8.2	CIM_Component (Backbone Switch to Fabric)	285
21.8.3	CIM_Component (IFR Switch to Fabric)	285
21.8.4	CIM_ComputerSystem (Backbone Switch)	286
21.8.5	CIM_ComputerSystem (IFR Switch)	287
21.8.6	CIM_FCActiveConnection	287
21.8.7	CIM_FCPort (IFR FCPort).....	288
21.8.8	CIM_ProtocolEndpoint	289
Annex A	(informative) SMI-S Information Model.....	290
Annex B	(informative) Structure of Fabric Profiles.....	291
Bibliography		311

Figure 1 - Experimental maturity level tag	26
Figure 2 - Implemented maturity level tag.....	27
Figure 3 - Stable maturity level tag	27
Figure 4 - Deprecated tag	28
Figure 5 - Fabric Instance	29
Figure 6 - NPIV Instance.....	31
Figure 7 - Zoning Instance (AdminDomain).....	32
Figure 8 - Zoning Instance (ComputerSystem).....	33
Figure 9 - Predefined Fabric IndicationFilters and Collection	34
Figure 10 - FDMI Instance	111
Figure 11 - Instance Diagram	121
Figure 12 - Specialization of Security Authorization Subprofile for Membership Policy.....	128
Figure 13 - Specialization of Security Authorization Subprofile for Connectivity Policy.....	129
Figure 14 - Fabric View Class Capabilities	135
Figure 15 - Topology View Class	136
Figure 16 - FCSwitch View Class	136
Figure 17 - RegisteredProfile/Subprofile, AdminDomain, and ComputerSystem Relationships.....	143
Figure 18 - Two Virtual Fabric and Two Partitioning Systems	144
Figure 19 - Two Virtual Fabrics and One Partitioning System	144
Figure 20 - Switch Instance Diagram.....	148
Figure 21 - Trunking Instance Diagram	149
Figure 22 - Predefined Switch IndicationFilters and Collection.....	152
Figure 23 - Switch Configuration Data Instance	195
Figure 24 - Switch Blade Instance	199
Figure 25 - Predefined Blades IndicationFilters and Collection	200
Figure 26 - Switch ComputerSystem and Partitioning System	211
Figure 27 - Switch and Partitioning System and Partitioning Ports.....	211
Figure 28 - Underlying System Port Settings and Capabilities	212
Figure 29 - FC Extender Node Instance	221
Figure 30 - FC Extender Group Instance.....	222
Figure 31 - Predefined Extender IndicationFilters and Collection.....	223
Figure 32 - SAS Expander Instance Diagram.....	265
Figure 33 - N Port Virtualizer	272
Figure 34 - IFR Switch Topology	282
Figure 35 - Inter Fabric Routing Ports.....	284
Figure B.1 - Structure of the Fabric Part Profiles	291
Figure B.2 - AdminDomain Properties	294
Figure B.3 - Associations on AdminDomains	296
Figure B.4 - Properties of various Switches	298
Figure B.5 - Associations on Switches	300
Figure B.6 - General ComputerSystem Properties	302
Figure B.7 - General ComputerSystem Associations	304
Figure B.8 - Switch FCPort Properties	306
Figure B.9 - Switch FCPort Associations	307
Figure B.10 - Non-Switch Port Properties	309
Figure B.11 - Non-Switch Port Associations	310

Table 1 - Supported Profiles for Fabric	35
Table 2 - Port OperationalStatus.....	36
Table 3 - OperationalStatus for ComputerSystem	36
Table 4 - CIM Elements for Fabric	41
Table 5 - SMI Referenced Properties/Methods for CIM_ActiveConnection	47
Table 6 - SMI Referenced Properties/Methods for CIM_AdminDomain (Fabric)	48
Table 7 - SMI Referenced Properties/Methods for CIM_AdminDomain (SAN).....	48
Table 8 - SMI Referenced Properties/Methods for CIM_Component (Platform to Fabric)	49
Table 9 - SMI Referenced Properties/Methods for CIM_Component (Switch to Fabric)	49
Table 10 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Host Platform).....	50
Table 11 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned Switch).....	50
Table 12 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Storage Platform).....	51
Table 13 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Switch).....	52
Table 14 - SMI Referenced Properties/Methods for CIM_ConnectivityCollection	53
Table 15 - SMI Referenced Properties/Methods for CIM_ContainedDomain	53
Table 16 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (Non-Switch to FCPort).....	53
Table 17 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (Switch to FCPort).....	54
Table 18 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ZoneCapabilities to Fabric).....	54
Table 19 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ZoneCapabilities to Switch)	54
Table 20 - SMI Referenced Properties/Methods for CIM_ElementSettingData (ZoneMembershipSettingData to Zone).....	55
Table 21 - SMI Referenced Properties/Methods for CIM_FCActiveConnection	55
Table 22 - SMI Referenced Properties/Methods for CIM_FCPort (Host FCPort)	56
Table 23 - SMI Referenced Properties/Methods for CIM_FCPort (Host NPIV FCPort)	57
Table 24 - SMI Referenced Properties/Methods for CIM_FCPort (Partitioned Switch FCPort)	58
Table 25 - SMI Referenced Properties/Methods for CIM_FCPort (Storage FCPort)	59
Table 26 - SMI Referenced Properties/Methods for CIM_FCPort (Switch FCPort)	60
Table 27 - SMI Referenced Properties/Methods for CIM_FilterCollection (Fabric Predefined FilterCollection)	61
Table 28 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint).....	61
Table 29 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)	62
Table 30 - SMI Referenced Properties/Methods for CIM_HostedCollection (Fabric to ConnectivityCollection)	62
Table 31 - SMI Referenced Properties/Methods for CIM_HostedCollection (Fabric to predefined FilterCollection).....	62
Table 32 - SMI Referenced Properties/Methods for CIM_HostedCollection (System to LogicalPortGroup).....	63
Table 33 - SMI Referenced Properties/Methods for CIM_HostedCollection (Zones or ZoneSets to Fabric)	63
Table 34 - SMI Referenced Properties/Methods for CIM_HostedCollection (Zones or ZoneSets to Switch)	63
Table 35 - SMI Referenced Properties/Methods for CIM_HostedDependency	64
Table 36 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric ComputerSystem Creation).....	64

Table 37 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric ComputerSystem Deletion).....	65
Table 38 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric ComputerSystem OperationalStatus).....	66
Table 39 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric FCPort Creation).....	67
Table 40 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric FCPort Deletion).....	68
Table 41 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric FCPort OperationalStatus)	69
Table 42 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Fabric Added or Removed).....	70
Table 43 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Fabric Merge or Segmentation).....	71
Table 44 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Switch Added or Removed).....	72
Table 45 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Switch Status Changed)	73
Table 46 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Zone Database Changed)	74
Table 47 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric ZoneSet Activated)	75
Table 48 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus).....	76
Table 49 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Fabric FCPort OperationalStatus)	77
Table 50 - SMI Referenced Properties/Methods for CIM_LogicalPortGroup	78
Table 51 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint).....	78
Table 52 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (LogicalPortGroup to FCPort)	78
Table 53 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Fabric Filters).....	79
Table 54 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (ZoneSet to Zone).....	79
Table 55 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint.....	79
Table 56 - SMI Referenced Properties/Methods for CIM_SystemDevice (Non-Switch FCPort to Fabric).....	80
Table 57 - SMI Referenced Properties/Methods for CIM_SystemDevice (Non-Switch FCPort to Platform)	80
Table 58 - SMI Referenced Properties/Methods for CIM_SystemDevice (Switch FCPort to Switch)	81
Table 59 - SMI Referenced Properties/Methods for CIM_Zone (Active).....	81
Table 60 - SMI Referenced Properties/Methods for CIM_Zone (Inactive).....	81
Table 61 - SMI Referenced Properties/Methods for CIM_ZoneCapabilities	82
Table 62 - SMI Referenced Properties/Methods for CIM_ZoneMembershipSettingData	83
Table 63 - SMI Referenced Properties/Methods for CIM_ZoneSet (Active)	84
Table 64 - SMI Referenced Properties/Methods for CIM_ZoneSet (Inactive).....	84
Table 65 - Supported Profiles for Enhanced Zoning and Enhanced Zoning Control	85
Table 66 - CIM Elements for Enhanced Zoning and Enhanced Zoning Control	89
Table 67 - SMI Referenced Properties/Methods for CIM_ElementSettingData (ZoneMembershipSettingData to NamedAddressCollection)	89
Table 68 - SMI Referenced Properties/Methods for CIM_HostedCollection (AdminDomain to Collection)	90
Table 69 - SMI Referenced Properties/Methods for CIM_HostedCollection (ComputerSystem to Collection).....	90
Table 70 - SMI Referenced Properties/Methods for CIM_MemberOfCollection	90

Table 71 - SMI Referenced Properties/Methods for CIM_NamedAddressCollection	91
Table 72 - SMI Referenced Properties/Methods for CIM_ZoneService (Zone Service)	91
Table 73 - CIM Elements for Zone Control	108
Table 74 - SMI Referenced Properties/Methods for CIM_HostedService (Fabric (AdminDomain) to ZoneService)	109
Table 75 - SMI Referenced Properties/Methods for CIM_HostedService (Switch (ComputerSystem) to ZoneService)	109
Table 76 - SMI Referenced Properties/Methods for CIM_ZoneService (Zone Service)	110
Table 77 - CIM Elements for FDMI	112
Table 78 - SMI Referenced Properties/Methods for CIM_Component (Host Server to Fabric)	113
Table 79 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Host Server)	113
Table 80 - SMI Referenced Properties/Methods for CIM_ControlledBy	113
Table 81 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity	114
Table 82 - SMI Referenced Properties/Methods for CIM_FCPort (Host FCPort)	114
Table 83 - SMI Referenced Properties/Methods for CIM_HostedCollection (System to LogicalPortGroup)	115
Table 84 - SMI Referenced Properties/Methods for CIM_InstalledSoftwareIdentity	115
Table 85 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (LogicalPortGroup to FCPort)	116
Table 86 - SMI Referenced Properties/Methods for CIM_PhysicalPackage (HBA Package)	116
Table 87 - SMI Referenced Properties/Methods for CIM_PortController	117
Table 88 - SMI Referenced Properties/Methods for CIM_Product (HBA Product)	117
Table 89 - SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent	117
Table 90 - SMI Referenced Properties/Methods for CIM_Realizes	118
Table 91 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Driver)	118
Table 92 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Firmware)	119
Table 93 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Option ROM)	119
Table 94 - SMI Referenced Properties/Methods for CIM_SystemDevice (ComputerSystem to FCPort)	120
Table 95 - SMI Referenced Properties/Methods for CIM_SystemDevice (ComputerSystem to PortController)	120
Table 96 - CIM Elements for Fabric Path Performance	122
Table 97 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData	123
Table 98 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe	123
Table 99 - SMI Referenced Properties/Methods for CIM_HostedCollection	123
Table 100 - SMI Referenced Properties/Methods for CIM_HostedNetworkPipe	124
Table 101 - SMI Referenced Properties/Methods for CIM_MemberOfCollection	124
Table 102 - SMI Referenced Properties/Methods for CIM_Network	124
Table 103 - SMI Referenced Properties/Methods for CIM_NetworkPipe	125
Table 104 - SMI Referenced Properties/Methods for CIM_NetworkPortStatistics	125
Table 105 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint	126
Table 106 - SMI Referenced Properties/Methods for CIM_StatisticsCollection	126
Table 107 - CIM Elements for FabricSecurity	130
Table 108 - SMI Referenced Properties/Methods for CIM_AuthorizationService	130
Table 109 - SMI Referenced Properties/Methods for CIM_AuthorizedPrivilege	131
Table 110 - SMI Referenced Properties/Methods for CIM_AuthorizedSubject	131
Table 111 - SMI Referenced Properties/Methods for CIM_AuthorizedTarget	131
Table 112 - SMI Referenced Properties/Methods for CIM_HostedService	132
Table 113 - SMI Referenced Properties/Methods for CIM_ServiceAffectsElement (ManagedElement to Service)	132
Table 114 - SMI Referenced Properties/Methods for CIM_ServiceAffectsElement (StorageHardwareID to Service)	132
Table 115 - SMI Referenced Properties/Methods for CIM_ServiceAvailableToElement	

(Fabric AdminDomain to Service)	133
Table 116 - SMI Referenced Properties/Methods for CIM_StorageHardwareID	133
Table 117 - CIM Elements for Fabric Views	137
Table 118 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (View Capabilities)	138
Table 119 - SMI Referenced Properties/Methods for SNIA_BaseInstance (View to FC Port)	138
Table 120 - SMI Referenced Properties/Methods for SNIA_BaseInstance (View to Switch)	139
Table 121 - SMI Referenced Properties/Methods for SNIA_ComponentView (FCSwitchView to Fabric)	139
Table 122 - SMI Referenced Properties/Methods for SNIA_ComponentView (TopologyView to Fabric)	140
Table 123 - SMI Referenced Properties/Methods for SNIA_FCSwitchView	140
Table 124 - SMI Referenced Properties/Methods for SNIA_TopologyView	141
Table 125 - SMI Referenced Properties/Methods for SNIA_ViewCapabilities	142
Table 126 - Supported Profiles for FabricVirtualFabrics	145
Table 127 - CIM Elements for FabricVirtualFabrics	145
Table 128 - SMI Referenced Properties/Methods for CIM_Component (AdminDomain to Partitioning CS)	146
Table 129 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile (SAN AdminDomain to Virtual Fabrics RegisteredProfile)	146
Table 130 - DetailedPortState for FCPort	150
Table 131 - PortAvailability for FCPort	150
Table 132 - Supported Profiles for Switch	153
Table 133 - CIM Elements for Switch	170
Table 134 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned Switch)	173
Table 135 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Switch)	174
Table 136 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage	175
Table 137 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (FCPort to FCPortCapabilities)	175
Table 138 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (System to FCSwitchCapabilities)	175
Table 139 - SMI Referenced Properties/Methods for CIM_ElementSettingData (FCPortSettings to FCPort)	176
Table 140 - SMI Referenced Properties/Methods for CIM_ElementSettingData (FCSwitchSettings to ComputerSystem)	176
Table 141 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPortRateStatistics to FCPort)	176
Table 142 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPortStatistics to FCPort)	177
Table 143 - SMI Referenced Properties/Methods for CIM_FCPort (Partitioned Switch FCPort)	177
Table 144 - SMI Referenced Properties/Methods for CIM_FCPort (Switch FCPort)	178
Table 145 - SMI Referenced Properties/Methods for CIM_FCPortCapabilities	180
Table 146 - SMI Referenced Properties/Methods for CIM_FCPortRateStatistics	180
Table 147 - SMI Referenced Properties/Methods for CIM_FCPortSettings	181
Table 148 - SMI Referenced Properties/Methods for CIM_FCPortStatistics	181
Table 149 - SMI Referenced Properties/Methods for CIM_FCSwitchCapabilities	183
Table 150 - SMI Referenced Properties/Methods for CIM_FCSwitchSettings	184
Table 151 - SMI Referenced Properties/Methods for CIM_FilterCollection (Switch Predefined FilterCollection)	185
Table 152 - SMI Referenced Properties/Methods for CIM_HostedCollection (Redundancy Set)	185
Table 153 - SMI Referenced Properties/Methods for CIM_HostedCollection (Statistics Collection)	185
Table 154 - SMI Referenced Properties/Methods for CIM_HostedCollection (Switch to predefined FilterCollection)	186

Table 155 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Switch ComputerSystem Creation).....	186
Table 156 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Switch ComputerSystem Deletion).....	187
Table 157 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Switch ComputerSystem OperationalStatus).....	188
Table 158 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Switch FCPort OperationalStatus).....	189
Table 159 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus).....	190
Table 160 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Switch FCPort OperationalStatus).....	191
Table 161 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (FCPort to RedundancySet).....	191
Table 162 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection).....	192
Table 163 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Switch Filters).....	192
Table 164 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint.....	192
Table 165 - SMI Referenced Properties/Methods for CIM_RedundancySet.....	193
Table 166 - SMI Referenced Properties/Methods for CIM_StatisticsCollection.....	193
Table 167 - SMI Referenced Properties/Methods for CIM_SystemDevice.....	194
Table 168 - CIM Elements for Switch Configuration Data.....	196
Table 169 - SMI Referenced Properties/Methods for CIM_ComputerSystem.....	197
Table 170 - SMI Referenced Properties/Methods for CIM_ConfigurationData.....	197
Table 171 - SMI Referenced Properties/Methods for CIM_ElementSettingData.....	198
Table 172 - CIM Elements for Blades.....	201
Table 173 - SMI Referenced Properties/Methods for CIM_FilterCollection (Blades Predefined FilterCollection).....	202
Table 174 - SMI Referenced Properties/Methods for CIM_HostedCollection (Switch to Blades predefined FilterCollection).....	203
Table 175 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Blades LogicalModule Creation).....	203
Table 176 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Blades LogicalModule Deletion).....	204
Table 177 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Blades LogicalModule OperationalStatus).....	205
Table 178 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Blades LogicalModule OperationalStatus).....	206
Table 179 - SMI Referenced Properties/Methods for CIM_LogicalModule.....	207
Table 180 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Blades FilterCollection to Switch FilterCollection).....	207
Table 181 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Blades Filters).....	208
Table 182 - SMI Referenced Properties/Methods for CIM_ModulePort.....	208
Table 183 - SMI Referenced Properties/Methods for CIM_PhysicalPackage (Logical Module).....	208
Table 184 - SMI Referenced Properties/Methods for CIM_Product (Blade).....	209
Table 185 - SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent.....	209
Table 186 - SMI Referenced Properties/Methods for CIM_Realizes (Logical Module Package).....	210
Table 187 - SMI Referenced Properties/Methods for CIM_SystemDevice (Logical Module).....	210
Table 188 - Supported Profiles for FabricSwitchPartitioning.....	212
Table 189 - CIM Elements for FabricSwitchPartitioning.....	213
Table 190 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned).....	214
Table 191 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioning).....	214
Table 192 - SMI Referenced Properties/Methods for CIM_ElementCapabilities	

(Association between NetworkPort and NetworkPortCapabilities)	215
Table 193 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile (Partitioning ComputerSystem to Switch Partitioning RegisteredProfile).....	215
Table 194 - SMI Referenced Properties/Methods for CIM_ElementSettingData (Association between NetworkPort and NetworkPortSettings)	215
Table 195 - SMI Referenced Properties/Methods for CIM_FCPort (Partitioned)	216
Table 196 - SMI Referenced Properties/Methods for CIM_HostedDependency (NetworkPort to FCPort)	217
Table 197 - SMI Referenced Properties/Methods for CIM_HostedDependency (Partitioning CS to Partitioned CS)	217
Table 198 - SMI Referenced Properties/Methods for CIM_NetworkPort (Partitioning).....	217
Table 199 - SMI Referenced Properties/Methods for CIM_NetworkPortCapabilities	218
Table 200 - SMI Referenced Properties/Methods for CIM_NetworkPortSettings	218
Table 201 - SMI Referenced Properties/Methods for CIM_SystemDevice (FCPort to Partitioned ComputerSystem)	219
Table 202 - SMI Referenced Properties/Methods for CIM_SystemDevice (NetworkPort to ComputerSystem)	219
Table 203 - Supported Profiles for Extender	224
Table 204 - CIM Elements for Extender	234
Table 205 - SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to PE)	237
Table 206 - SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to RemoteSAP)	238
Table 207 - SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to RemoteSAP)	238
Table 208 - SMI Referenced Properties/Methods for CIM_BindsTo (PE to RemotePort).....	238
Table 209 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to IPPE)	239
Table 210 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to PE).....	239
Table 211 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to RemotePort)	239
Table 212 - SMI Referenced Properties/Methods for CIM_Component	240
Table 213 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Extender Node).....	240
Table 214 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation	240
Table 215 - SMI Referenced Properties/Methods for CIM_ElementSettingData (IPPE to IPSettings).....	241
Table 216 - SMI Referenced Properties/Methods for CIM_ElementSettingData (PE to FCIPSettings).....	241
Table 217 - SMI Referenced Properties/Methods for CIM_ElementSettingData (System to FCIPSettings)	242
Table 218 - SMI Referenced Properties/Methods for CIM_ElementSettingData (System to IPSettings)	242
Table 219 - SMI Referenced Properties/Methods for CIM_ElementSettingData (System to TCPSettings)	242
Table 220 - SMI Referenced Properties/Methods for CIM_ElementSettingData (TCPPE to TCPSettings)	243
Table 221 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics).....	243
Table 222 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPort to FCPortStatistics)	243
Table 223 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (IPPE to IPEndpointStatistics)	244
Table 224 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (System to TCPStatisticalData)	244
Table 225 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics).....	244
Table 226 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (PE to NetworkPipe)	245
Table 227 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe	

(TCPPE to NetworkPipe)	245
Table 228 - SMI Referenced Properties/Methods for CIM_EthernetPort	245
Table 229 - SMI Referenced Properties/Methods for CIM_EthernetPortStatistics	246
Table 230 - SMI Referenced Properties/Methods for CIM_FCIPSettings	247
Table 231 - SMI Referenced Properties/Methods for CIM_FCPort	247
Table 232 - SMI Referenced Properties/Methods for CIM_FCPortStatistics	248
Table 233 - SMI Referenced Properties/Methods for CIM_FilterCollection (Extender Predefined FilterCollection)	250
Table 234 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)	250
Table 235 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint)	251
Table 236 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)	251
Table 237 - SMI Referenced Properties/Methods for CIM_HostedCollection (Extender to predefined FilterCollection)	251
Table 238 - SMI Referenced Properties/Methods for CIM_HostedNetworkPipe	252
Table 239 - SMI Referenced Properties/Methods for CIM_IPEndpointStatistics	252
Table 240 - SMI Referenced Properties/Methods for CIM_IPProtocolEndpoint	253
Table 241 - SMI Referenced Properties/Methods for CIM_IPSettings	253
Table 242 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Extender ComputerSystem Creation)	254
Table 243 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Extender ComputerSystem Deletion)	255
Table 244 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Extender ComputerSystem OperationalStatus)	256
Table 245 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Extender ComputerSystem OperationalStatus)	257
Table 246 - SMI Referenced Properties/Methods for CIM_LANEndpoint	257
Table 247 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Extender Filters)	258
Table 248 - SMI Referenced Properties/Methods for CIM_Network	258
Table 249 - SMI Referenced Properties/Methods for CIM_NetworkPipe	259
Table 250 - SMI Referenced Properties/Methods for CIM_NetworkPipeComposition	259
Table 251 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint	259
Table 252 - SMI Referenced Properties/Methods for CIM_RemotePort	260
Table 253 - SMI Referenced Properties/Methods for CIM_RemoteServiceAccessPoint	260
Table 254 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to EthernetPort)	261
Table 255 - SMI Referenced Properties/Methods for CIM_SystemDevice (System to FCPort)	261
Table 256 - SMI Referenced Properties/Methods for CIM_TCPEndpointStatistics	261
Table 257 - SMI Referenced Properties/Methods for CIM_TCPProtocolEndpoint	262
Table 258 - SMI Referenced Properties/Methods for CIM_TCPSettings	262
Table 259 - SMI Referenced Properties/Methods for CIM_TCPStatisticalData	263
Table 260 - Supported Profiles for SAS Expander	266
Table 261 - CIM Elements for SAS Expander	267
Table 262 - SMI Referenced Properties/Methods for CIM_ComputerSystem	267
Table 263 - SMI Referenced Properties/Methods for CIM_ConcreteComponent	268
Table 264 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity	268
Table 265 - SMI Referenced Properties/Methods for CIM_Product	269
Table 266 - SMI Referenced Properties/Methods for CIM_ProductElementComponent	269
Table 267 - SMI Referenced Properties/Methods for CIM_SASPort	269
Table 268 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity	270
Table 269 - SMI Referenced Properties/Methods for CIM_SystemDevice (Port)	270

Table 270 - SMI Referenced Properties/Methods for CIM_SystemDevice (SASPHY)	270
Table 271 - SMI Referenced Properties/Methods for SNIA_SASPHY	271
Table 272 - Supported Profiles for N Port Virtualizer	273
Table 273 - CIM Elements for N Port Virtualizer	273
Table 274 - SMI Referenced Properties/Methods for CIM_Component (N Port Virtualizer to Fabric)	274
Table 275 - SMI Referenced Properties/Methods for CIM_ComputerSystem (N Port Virtualizer)	275
Table 276 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage (N Port Virtualizer to Physical Package)	275
Table 277 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (ProtocolEndpoint to Gateway FCPort)	276
Table 278 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (ProtocolEndpoint to NPIV FCPort)	276
Table 279 - SMI Referenced Properties/Methods for CIM_FCActiveConnection (Gateway)	276
Table 280 - SMI Referenced Properties/Methods for CIM_FCActiveConnection (N Port Virtualization)	277
Table 281 - SMI Referenced Properties/Methods for CIM_FCPort (Fabric NPIV)	277
Table 282 - SMI Referenced Properties/Methods for CIM_FCPort (Gateway)	278
Table 283 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (N Port Virtualizer System to ProtocolEndpoint)	279
Table 284 - SMI Referenced Properties/Methods for CIM_LogicalIdentity (NPIV Port to Switch Port)	280
Table 285 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint (N Port Virtualizer)	280
Table 286 - SMI Referenced Properties/Methods for CIM_SystemDevice (N Port Virtualizer Gateway FCPort to Gateway System)	280
Table 287 - SMI Referenced Properties/Methods for CIM_SystemDevice (N Port Virtualizer NPIV FCPort to Gateway System)	281
Table 288 - CIM Elements for Inter Fabric Routing	285
Table 289 - SMI Referenced Properties/Methods for CIM_Component (Backbone Switch to Fabric)	285
Table 290 - SMI Referenced Properties/Methods for CIM_Component (IFR Switch to Fabric)	286
Table 291 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Backbone Switch)	286
Table 292 - SMI Referenced Properties/Methods for CIM_ComputerSystem (IFR Switch)	287
Table 293 - SMI Referenced Properties/Methods for CIM_FCActiveConnection	288
Table 294 - SMI Referenced Properties/Methods for CIM_FCPort (IFR FCPort)	288
Table 295 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint	289
Table B.1 - AdminDomain Properties	293
Table B.2 - AdminDomain Associations	294
Table B.3 - Switch ComputerSystem Properties	297
Table B.4 - Switch ComputerSystem Associations	298
Table B.5 - Non-Switch ComputerSystem Properties	301
Table B.6 - Non-Switch ComputerSystem Associations	302
Table B.7 - Switch FCPort Properties	304
Table B.8 - Switch FCPort Associations	306
Table B.9 - Non-Switch Port Properties	308
Table B.10 - Non-Switch Port Associations	309

INFORMATION TECHNOLOGY – STORAGE MANAGEMENT –

Part 6: Fabric

FOREWORD

- 1) ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.
- 2) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees and ISO member bodies.
- 3) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC National Committees and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO, IEC or ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.
- 5) ISO and IEC do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. ISO or IEC are not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC National Committees or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.
- 8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this ISO/IEC publication may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 24775-6 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This International Standard, together with ISO/IEC 24775-1 to ISO/IEC 24775-5 and ISO/IEC 24775-7 and ISO/IEC 24775-8, replaces ISO/IEC 24775, second edition, published in 2011, and constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- a) reorganization into eight parts;
- b) maturity identification using stages; and
- c) new profiles.

The list of all currently available parts of the ISO/IEC 24775 series, under the general title *Information technology – Storage management*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IMPORTANT – The “colour inside” logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24775-6:2014

Introduction

This part of ISO/IEC 24775 defines management profiles for Autonomous (top level) profiles for programs and devices whose central function is providing support for storage networking. The standard includes fabric management including topology and device management for switches. This standard also provides management of extenders that pass fibre channel frames over other protocols as well as a gateway that maps and translates iSCSI to Fibre Channel. As part of fabric management, this International Standard also has controls for fibre channel zoning and fibre channel security.

Parts of this Standard

This International Standard is subdivided into the following parts:

- *Information technology – Storage management – Part 1: Overview*
- *Information technology – Storage management – Part 2: Common architecture*
- *Information technology – Storage management – Part 3: Common profiles*
- *Information technology – Storage management – Part 4: Block devices*
- *Information technology – Storage management – Part 5: Filesystems*
- *Information technology – Storage management – Part 6: Fabric*
- *Information technology – Storage management – Part 7: Host elements*
- *Information technology – Storage management – Part 8: Media libraries*

Changes since the Last Edition

The changes since ISO/IEC 24775:2011 fall into three broad categories:

- **Improved organization.** The stand-alone standard has been split into eight parts to provide more information more easily. The parts are:
 - *Part 1 Overview:* The overview book provides a high level overview of the standard.
 - *Part 2 Common Architecture:* This part covers general information about the interface, such as security and protocols.
 - *Part 3 Common Profiles:* This part covers component profiles that extend profiles in other books, such as target ports and job control.
 - *Part 4 Block Devices:* This part covers storage profiles that support various forms of disk storage.
 - *Part 5 Filesystems:* This part covers profiles that support filesystems, such as NAS (Network Attached Storage).
 - *Part 6 Fabric:* This part covers profiles that deal with interconnection of host servers and storage devices, such as switches.
 - *Part 7 Host Elements:* This part covers profiles for storage software on host servers, such as disk partitioning and Host Hardware RAID controllers.
 - *Part 8 Media Libraries:* This part covers profiles that deal with removable media such as tape libraries.
- **Maturity identification.** As material is added to the standard it goes through various stages of maturity. The initial stage is *Experimental*, which is material that has not yet been implemented and is subject to change. The other stages indicate the degree of implementations. The stages are:
 - **Experimental:** Full design review, no commercial implementations.
 - **Implemented:** Initial implementations available, may be removed at minor revision.
 - **Stable:** Three or more vendors have implemented the identified material, backward compatibility assured, removed only at major revision.

- **Finalized:** Relies solely on Finalized content, deprecated only at major revision.
- **Deprecated:** Obsolete material, may be removed in future revisions.
For a more detailed explanation of each maturity level and its typographical indication, see Clause 4 Typographical Conventions.
- **Expanded scope.** The range of SAN components modeled by the profiles defined in the parts has been greatly expanded.
 - New profiles include:
 - *Part 3 Common Profiles:* Serial Attached SCSI (SAS) Target Port, Serial ATA (SATA) Target Ports, SB Target Port, SAS Initiator Ports, ATA Initiator Ports, FC-SB-x Initiator Ports, FCoE Initiator Ports, Power Supply, Fan, Sensors, Base Server, Media Access Device, Storage Enclosure, Software Inventory, Profile Registration, Proxy Server System Management, Operational Power.
 - *Part 4 Block Devices:* Block Storage Views, CKD Block Services, Erasure, Storage Server Asymmetry, Volume Composition, Storage Element Protection, Replication Services, Pools from Volumes, Group Masking and Mapping, Thin Provisioning.
 - *Part 5 Filesystems:* File Export, File Server Manipulation, File Storage, Filesystem, Filesystem Copy Services, Filesystem Performance, Filesystem Quotas, NAS Network Port, Host Filesystem, Filesystem Remote Copy Services.
 - *Part 6 Fabric:* Fibre Channel Security, Fabric Views, Virtual Fabrics, Switch Partitioning, SAS Expander, N Port Virtualizer, Inter Fabric Routing.
 - *Part 7 Host Elements:* Storage HBA, Host Hardware RAID Controller.
 - *Part 8 Media Libraries:* Partitioned Tape Library, Virtual Tape Library, Virtual Tape Library Copy and Library Views.
 - The following experimental profiles were removed from the International Standard:
 - *Part 3 Common Profiles:* Security, 3rd Party Authentication, Authorization, Credential Management, Identity Management, Security Role Based Access Control and Security Resource Ownership.
 - *Part 4 Block Devices:* Pool Management Policy.
 - The following profiles were deprecated:
 - *Part 3 Common Profiles:* Cascading (replaced by direct use of cascading classes).
 - *Part 4 Block Devices:* Volume Management (and not replaced).
 - *Part 6 Fabric:* Router (and not replaced).
 - *Part 7 Host Elements:* FC HBA (replaced by Storage HBA), SB Multipath Management (and not replaced).
 - *Part 8 Media Libraries:* InterLibraryPort Connection (and not replaced).
 - In addition, many of the existing profiles have been enhanced.

This International Standard was prepared by the SNIA (Storage Networking Industry Association)¹. The standard is often referred to as **SMI-S** (*Storage Management Initiative Specification*).

1. Storage Networking Industry Association, 425 Market Street, Suite 1020, San Francisco, CA 94105, U.S.A., <http://www.snia.org>

INFORMATION TECHNOLOGY – STORAGE MANAGEMENT – Part 6: Fabric

1 Scope

This part of ISO/IEC 24775 defines management profiles for Autonomous (top level) profiles for programs and devices whose central function is providing support for storage networking. This standard includes four autonomous profiles:

- Fabric

This profile defines the model and functions of a storage network including topology and zoning control.
- Switch

This profile defines the model and functions of a Fibre Channel Switch including state, status, and control of the device and its connections and product information,
- Extender

This profile defines the model and functions of a networking device that allows for fibre channel to be extended over other networks, and specifically over IP (FCIP).
- iSCSI to FC Gateway

This profile defines the model and functions of a networking device that maps and switches iSCSI frames from an IP network to a fibre channel fabric.

2 Normative references

The following documents, in whole or in part, are normatively referenced in the document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document, including any amendments, applies.

ISO/IEC 24775-2:2014, *Information technology – Storage management – Part 2: Common architecture*

ISO/IEC 24775-3:2014, *Information technology – Storage management – Part 3: Common profiles*

ISO/IEC 24775-7:2014, *Information technology – Storage management – Part 7: Host elements*

ISO/IEC 24775-8:2014, *Information technology – Storage management – Part 8: Media libraries*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 24775-2, apply.

4 Typographical conventions

4.1 Maturity model

In addition to informative and normative content, this International Standard includes guidance about the maturity of emerging material that has completed a rigorous design review but has limited implementation in commercial products. This material is clearly delineated as described in Clause 4. The typographical convention is intended to provide a sense of the maturity of the affected material, without altering its normative content. By recognizing the relative maturity of different clauses of the International Standard, an implementer should be able to make more informed decisions about the adoption and deployment of different portions of the International Standard in a commercial product.

This International Standard has been structured to convey both the formal requirements and assumptions of the Information Technology – Storage Management API and its emerging implementation and deployment lifecycle. Over time, the intent is that all content in the International Standard will represent a mature and stable design, be verified by extensive implementation experience, assure consistent support for backward compatibility, and rely solely on content material that has reached a similar level of maturity. Unless explicitly labeled with one of the subordinate maturity levels defined for this International Standard, content is assumed to satisfy these requirements and is referred to as “Finalized”. Since much of the evolving International Standard content in any given release will not have matured to that level, this International Standard defines three subordinate levels of implementation maturity that identify important aspects of the content’s increasing maturity and stability. Each subordinate maturity level is defined by its level of implementation experience, its stability and its reliance on other emerging standards. Each subordinate maturity level is identified by a unique typographical tagging convention that clearly distinguishes content at one maturity model from content at another level.

4.2 Experimental maturity level

No material is included in this International Standard unless its initial architecture has been completed and reviewed. Some content included in this International Standard has complete and reviewed design, but lacks implementation experience and the maturity gained through implementation experience. This content is included in order to gain wider review and to gain implementation experience. This material is referred to as “Experimental”. It is presented here as an aid to implementers who are interested in likely future developments within the SMI International Standards. The contents of an Experimental profile may change as implementation experience is gained. There is a high likelihood that the changed content will be included in an upcoming revision of the International Standard. Experimental material can advance to a higher maturity level as soon as implementations are available. Figure 1 is a sample of the typographical convention for Experimental content.

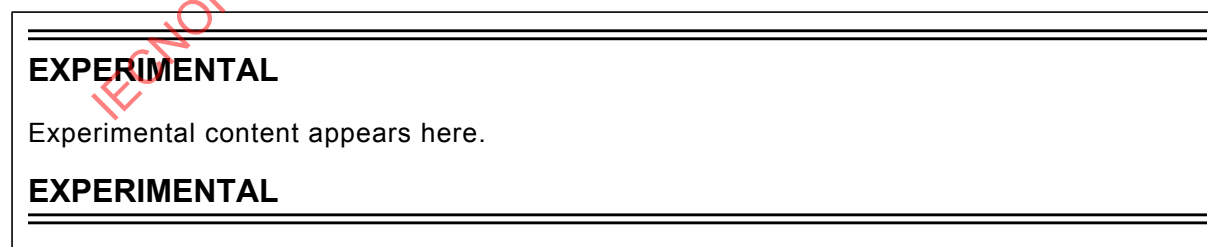


Figure 1 - Experimental maturity level tag

4.3 Implemented maturity level

Profiles for which initial implementations have been completed are classified as “Implemented”. This indicates that at least two different vendors have implemented the profile, including at least one provider implementation. At this maturity level, the underlying architecture and modeling are stable, and changes in future revisions will be limited to the correction of deficiencies identified through additional implementation experience. Should the material become obsolete in the future, it shall be deprecated in a minor revision of the

International Standard prior to its removal from subsequent releases. Figure 2 is a sample of the typographical convention for Implemented content.

IMPLEMENTED
Implemented content appears here.
IMPLEMENTED

Figure 2 - Implemented maturity level tag

4.4 Stable maturity level

Once content at the Implemented maturity level has garnered additional implementation experience, it can be tagged at the Stable maturity level. Material at this maturity level has been implemented by three different vendors, including both a provider and a client. Should material that has reached this maturity level become obsolete, it may only be deprecated as part of a minor revision to the International Standard. Material at this maturity level that has been deprecated may only be removed from the International Standard as part of a major revision. A profile that has reached this maturity level is guaranteed to preserve backward compatibility from one minor International Standard revision to the next. As a result, Profiles at or above the Stable maturity level shall not rely on any content that is Experimental. Figure 3 is a sample of the typographical convention for Implemented content.

STABLE
Stable content appears here.
STABLE

Figure 3 - Stable maturity level tag

4.5 Finalized maturity level

Content that has reached the highest maturity level is referred to as "Finalized." In addition to satisfying the requirements for the Stable maturity level, content at the Finalized maturity level shall solely depend upon or refine material that has also reached the Finalized level. If International Standard content depends upon material that is not under the control of the SNIA, and therefore not subject to its maturity level definitions, then the external content is evaluated by the SNIA to assure that it has achieved a comparable level of completion, stability, and implementation experience. Should material that has reached this maturity level become obsolete, it may only be deprecated as part of a major revision to the International Standard. A profile that has reached this maturity level is guaranteed to preserve backward compatibility from one minor International Standard revision to the next. Over time, it is hoped that all International Standard content will attain this maturity level. Accordingly, there is no special typographical convention, as there is with the other, subordinate maturity levels. Unless content in the International Standard is marked with one of the typographical conventions defined for the subordinate maturity levels, it should be assumed to have reached the Finalized maturity level.

4.6 Deprecated Material

Non-Experimental material can be deprecated in a subsequent revision of the International Standard. Sections identified as "Deprecated" contain material that is obsolete and not recommended for use in new development efforts. Existing and new implementations may still use this material, but shall move to the newer approach as soon as possible. The maturity level

of the material being deprecated determines how long it will continue to appear in the International Standard. Implemented content shall be retained at least until the next revision of the specialization, while Stable and Finalized material shall be retained until the next major revision of the International Standard. Providers shall implement the deprecated elements as long as it appears in the International Standard in order to achieve backward compatibility. Clients may rely on deprecated elements, but are encouraged to use non-deprecated alternatives when possible.

Deprecated sections are documented with a reference to the last published version to include the deprecated section as normative material and to the section in the current International Standard with the replacement. Figure 4 contains a sample of the typographical convention for deprecated content.



Figure 4 - Deprecated tag

STABLE

5 Fabric Profile

5.1 Description

5.1.1 General

Figure 5 illustrates the basic elements of the Fabric Profile.

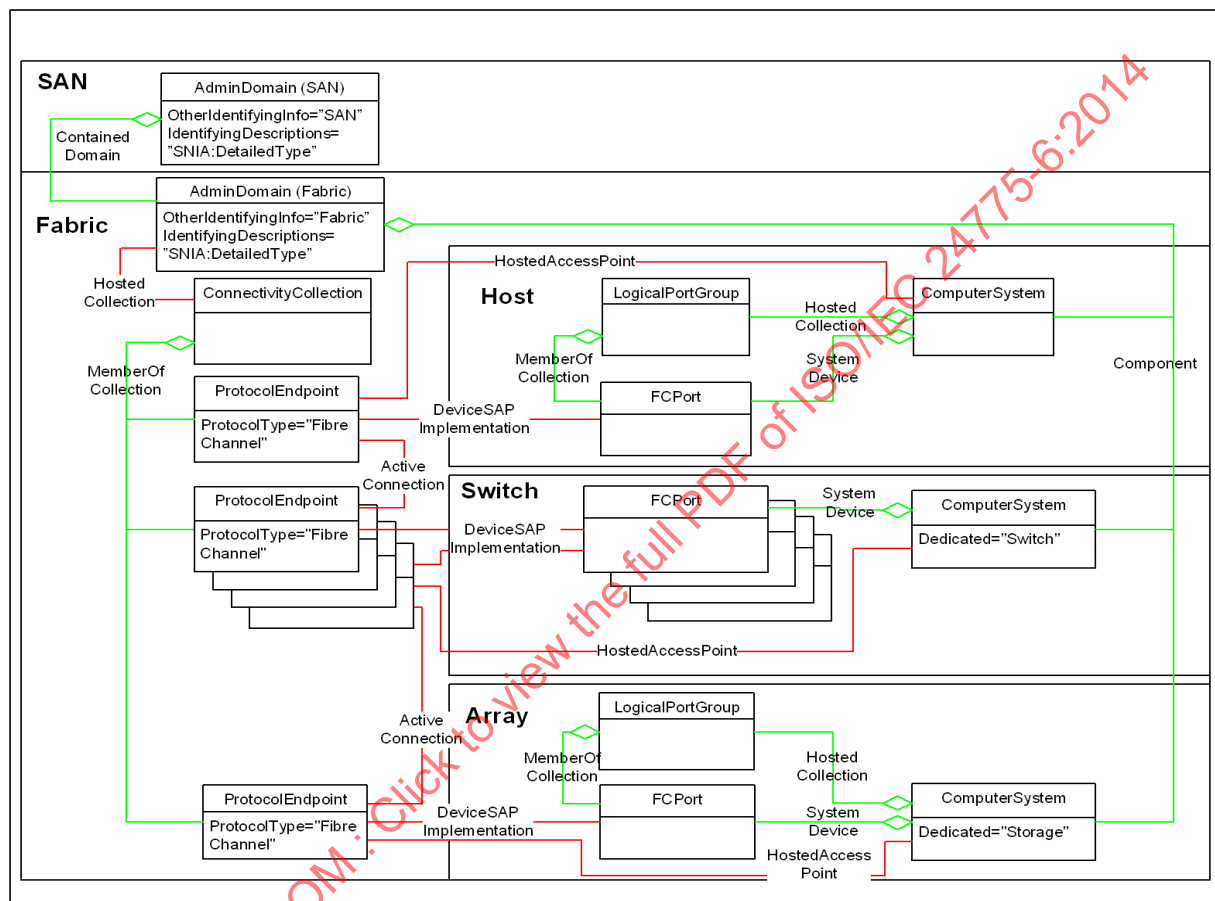


Figure 5 - Fabric Instance

5.1.2 SANs and Fabrics as AdminDomains

A SAN and Fabric are represented in CIM by AdminDomain. A SAN contains one or more Fabrics, which are modeled as AdminDomains. The "containment" of Fabrics to SANs is through the association ContainedDomain. AdminDomain is sub-classed from System. This is significant because a SAN and a Fabric can be considered a group of components that operate together as a single system and should be/are managed as such. The relationship of the Fabrics in a SAN is intended to be able to aggregate all the Fabrics that are in the same physical topology. The possible use cases include: Fibre Channel Virtual Fabrics which have multiple Virtual Fabrics in the same underlying physical topology, Fibre Channel Inter-Fabric Routing which routes Fibre Channel frames between Fabrics, and redundant Fabrics where two physically separate networks connect the same set of hosts and storage.

An AdminDomain in CIM is keyed by the property Name with an associated optional property NameFormat. The naming convention for SANs are left up to the implementation, but the Name shall be correlatable (the same) when multiple agents are managing the same SAN for Virtual Fabric (VF) and Inter-Fabric Routing (IFR).

For Fibre Channel Fabrics, the identifier (AdminDomain.Name) is the Fabric WWN as reported by the fibre channel fabric services. The AdminDomain for the Fibre Channel Fabric shall have a NameFormat of WWN.

5.1.3 Fabrics and Topology

A Fabric in CIM minimally contains a ConnectivityCollection and its component systems. They are associated to the Fabric by the association Component. For the purposes of this discussion, it is assumed one models both.

ConnectivityCollection represents the foundation necessary for routing (and is the reason it is defined in the Network model). A ConnectivityCollection groups a set of ProtocolEndpoints that are able to communicate with each other directly. The ProtocolEndpoint is associated to the ConnectivityCollection by MemberOfCollection. A link is represented by the association ActiveConnection, which associates two ProtocolEndpoints, defined as a connection that is currently carrying traffic or is configured to carry traffic.

It is important at this point to clarify the relationship (or use) of the ProtocolEndpoint versus the use of FCPort (discussed later). A NetworkPort (from which FCPort is subclassed) is the device that is used to represent the logical aspects of the link and data layers. The ProtocolEndpoint is used to represent the higher network layers for routing. This is best understood when thinking about Ethernet and IP, but applies to fibre channel also. When two ProtocolEndpoints are capable of communicating, the association ActiveConnection is used to represent the capability to communicate and completes the picture of the topology.

One can ultimately represent multiple ConnectivityCollections (e.g., FC, IP (over FC), and IP (FC encapsulated in IP) for the same fibre channel fabric.

The minimum set of requirements for this profile is based on FC-GS-6.

5.1.4 Systems and NetworkPorts

As discussed in 5.1.3, a Port is associated to a device to represent the link layer. A NetworkPort is associated to the ProtocolEndpoint by DeviceSAPImplementation and “joins” the System and Device model to the Network model. Instantiation of DeviceSAPImplementation, ProtocolEndpoint, and ActiveConnection is not necessary if the transceiver is not installed or the cable connecting the port to another port is not installed since the device is not capable of communicating.

Systems, or in this case ComputerSystem, represent the fabric elements that contain Ports. These are typically Hosts, Switches and Storage Systems. In Fibre Channel, these are called Platforms and Interconnect Elements. The property Dedicated in ComputerSystem allows these fabric elements to be identified. For a host, Dedicated is set to “Not Dedicated”, for a switch, Dedicated is set to “Switch”, and for a storage system, Dedicated is set to “Storage”. The Ports on a System are associated by SystemDevice.

Discovery from the viewpoint of the fabric includes the end device, but often times the information available is minimal or not available. In the case of Fibre Channel, this occurs if the platform database is not populated. If this is the case, then discovery cannot tell whether a Fibre Channel Node is contained within the same platform or not. When this occurs, ComputerSystem is not instantiated and the LogicalPortGroup representing the Node and the FCPort are associated to the AdminDomain representing the Fabric.

The instrumentation needs to respond to physical fabric changes by adding or removing Logical elements to the AdminDomain. Adding an element to the fabric is straightforward. However, it is not always clear when an element has been removed. The device may have been reset, or temporarily shut down, in which case it would be an element in the fabric with an “unknown” status. The lifetime of objects that can no longer be discovered is implementation specific.

If the instrumentation is unable to determine the type of platform discovered (defined in FC-GS-6), then the agent shall set the ComputerSystem.Dedicated property to “Unknown”.

Additional identification information about Switch ComputerSystem (e.g., DomainID) is placed in OtherIdentifyingInfo property.

EXPERIMENTAL

5.1.5 N Port Virtualization (NPIV)

Within a fabric, a host or an N Port Virtualizer can "virtualize" one or more fibre channel ports for each "physical" fibre channel port, known as NPIV. To model NPIV, Figure 6 is extended as shown in Figure 6 with multiple ActiveConnections to the same switch port. The association, HostedDependency, is used to associate the FCPorts representing the "physical" fibre channel port or hosting FCPort with the "virtual" fibre channel ports with the "virtual" port or dependent hosted FCPort.

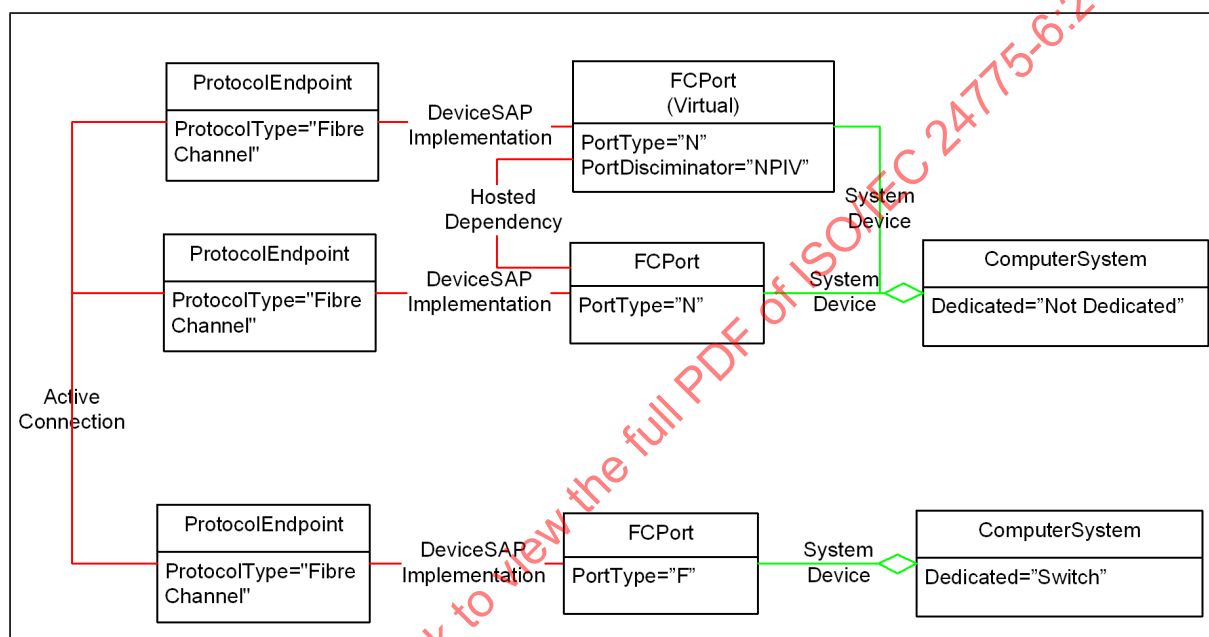


Figure 6 - NPIV Instance

EXPERIMENTAL

5.1.6 Zoning

Figure 7 illustrates the basic model for zoning.

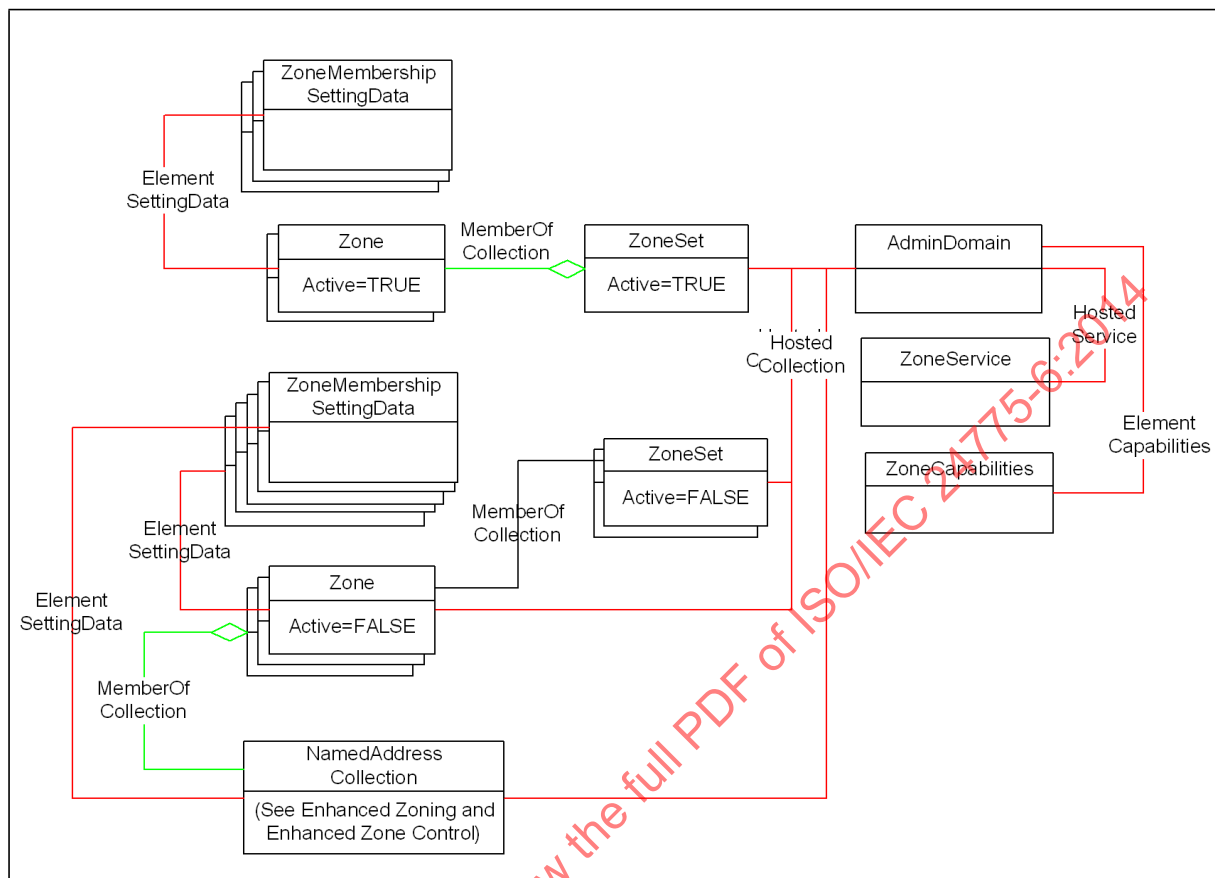


Figure 7 - Zoning Instance (AdminDomain)

The zoning model is based on FC-GS-4. This model represents the management model for defining Zone Sets, Zones, and Zone Members and “activation” of a Zone Set for a fabric. In the following discussion it may be helpful to also define the following:

- **Active ZoneSet**: the Zone Set currently enforced by the Fabric.
- **Zone Set Database**: The database of the Zone Sets not enforced by the Fabric. Referred to in this document as the Inactive Zone Sets.
- **Zoning Definitions**: a generic term used to indicate both the above concepts.

The zoning model refers to a Zone Set as ZoneSet, a Zone as Zone, ZoneAlias as a NamedAddressCollection, and Zone Member as ZoneMembershipSettingData. ZoneSets shall only contain Zones associated by MemberOfCollection. Zones shall only contain ZoneMembershipSettingData associated by ElementSettingData or NamedAddressCollections associated by MemberOfCollection. For more information with regards to NamedAddressCollection, see Clause 6 Enhanced Zoning and Enhanced Zone Control Subprofile.

The class ZoneMembershipSettingData has two properties that indicate how the device was identified to be “zoned”. They are ConnectivityMemberType (e.g., PermanentAddress for WWN, NetworkAddress for FCID, etc.) and ConnectivityMemberID which contains the actual device identifier.

The Active Zone Set, defined by an instance of ZoneSet with the Active property set to TRUE, shall only be hosted on the AdminDomain representing the Fabric. The Inactive Zone Sets, defined by an instance of ZoneSet with the Active property set to FALSE, shall be hosted on either the AdminDomain representing the Fabric as shown in Figure 7 or the ComputerSystem representing the switch as shown

in Figure 8. It is allowed to have no ZoneSets (active or inactive), only an active ZoneSet, only an inactive ZoneSet(s), or both an inactive ZoneSet(s) and an active ZoneSet.

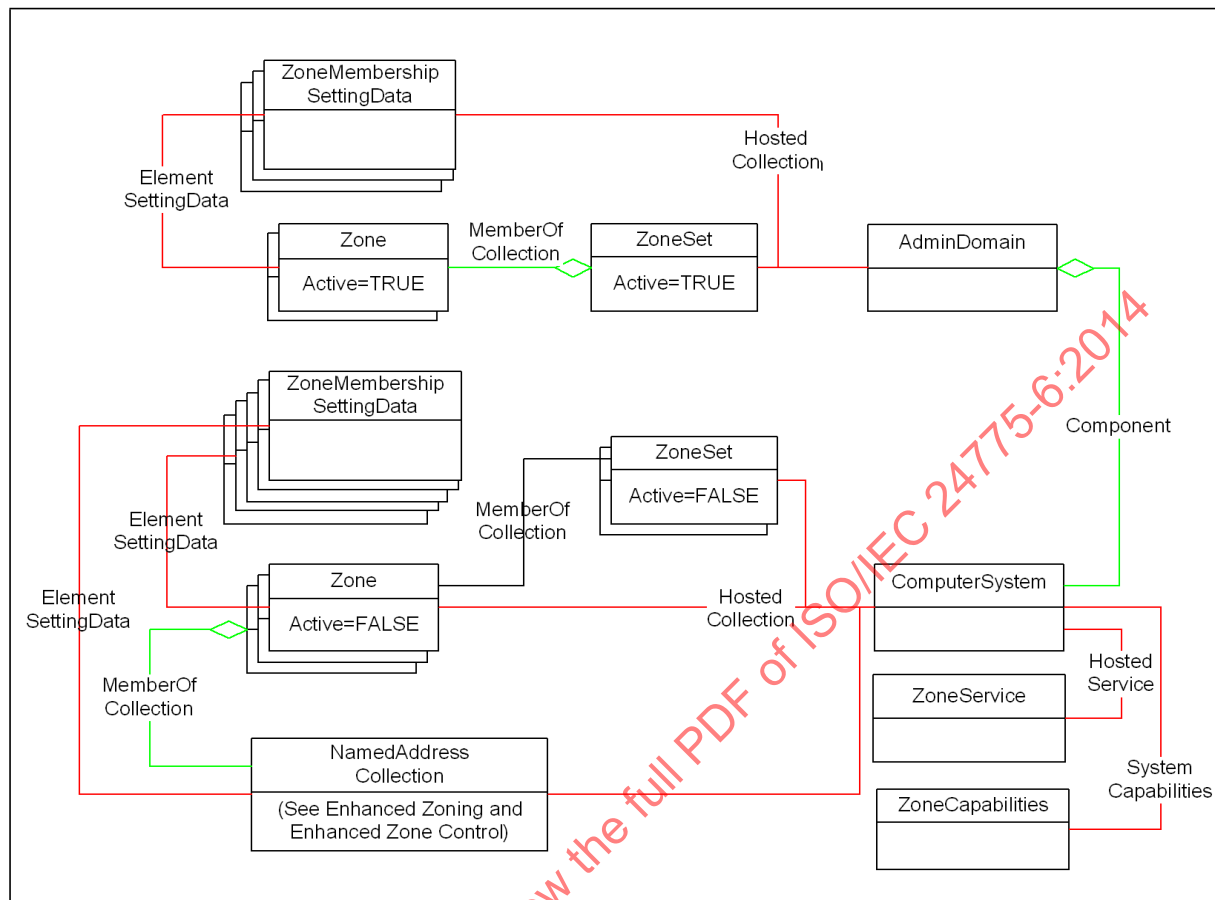


Figure 8 - Zoning Instance (ComputerSystem)

The ZoneService and ZoneCapabilities are also associated to the same System (AdminDomain or ComputerSystem) as the Inactive Zone Sets using the association HostedService or ElementCapabilities, respectively.

ZoneService provides the configuration methods to control create ZoneSets, Zones, Zone Aliases, and Zone Members, as well as activation of the Zone Set. This service and its methods are described in the Clause 6 Enhanced Zoning and Enhanced Zone Control Subprofile.

EXPERIMENTAL

5.1.7 Conditional Classes and Properties

Table 4 contains some classes which are conditional on implementation of other profiles in the Fabric Book:

- CIM_ComputerSystem (Partitioned Switch) is conditional on the implementation of the Virtual Fabrics component profile;
- CIM_FCPort (Partitioned Switch FCPort) is conditional on the implementation of the Switch Partitioning component profile.

In addition, some of the mandatory classes of the Fabric Profile include properties that are conditional on the implementation of other profiles in the Fabric book.

- Some properties in CIM_ComputerSystem (Switch) are conditional on the implementation of the Switch Profile.
- Some properties in CIM_FCPort (Switch FCPort) are conditional on the implementation of the Switch Profile.

EXPERIMENTAL

EXPERIMENTAL

5.1.8 Fabric Predefined Indications

If the optional Experimental Indication profile is supported by an implementation of the Fabric profile, it shall implement the SNIA_IndicationConfigurationService and its associated SNIA_IndicationConfigurationCapabilities. If the implementation supports predefined IndicationFilters or predefined IndicationFilterCollections this shall be indicated in the SupportedFeatures property of the SNIA_IndicationConfigurationCapabilities. If a value "3" is present, it means the implementation supports predefined IndicationFilters. If a value of "5" is present, it means the implementation supports predefined IndicationFilterCollections.

Figure 9 illustrates classes that shall be populated by the Fabric profile if both "3" and "5" are present in the SupportedFeatures property.

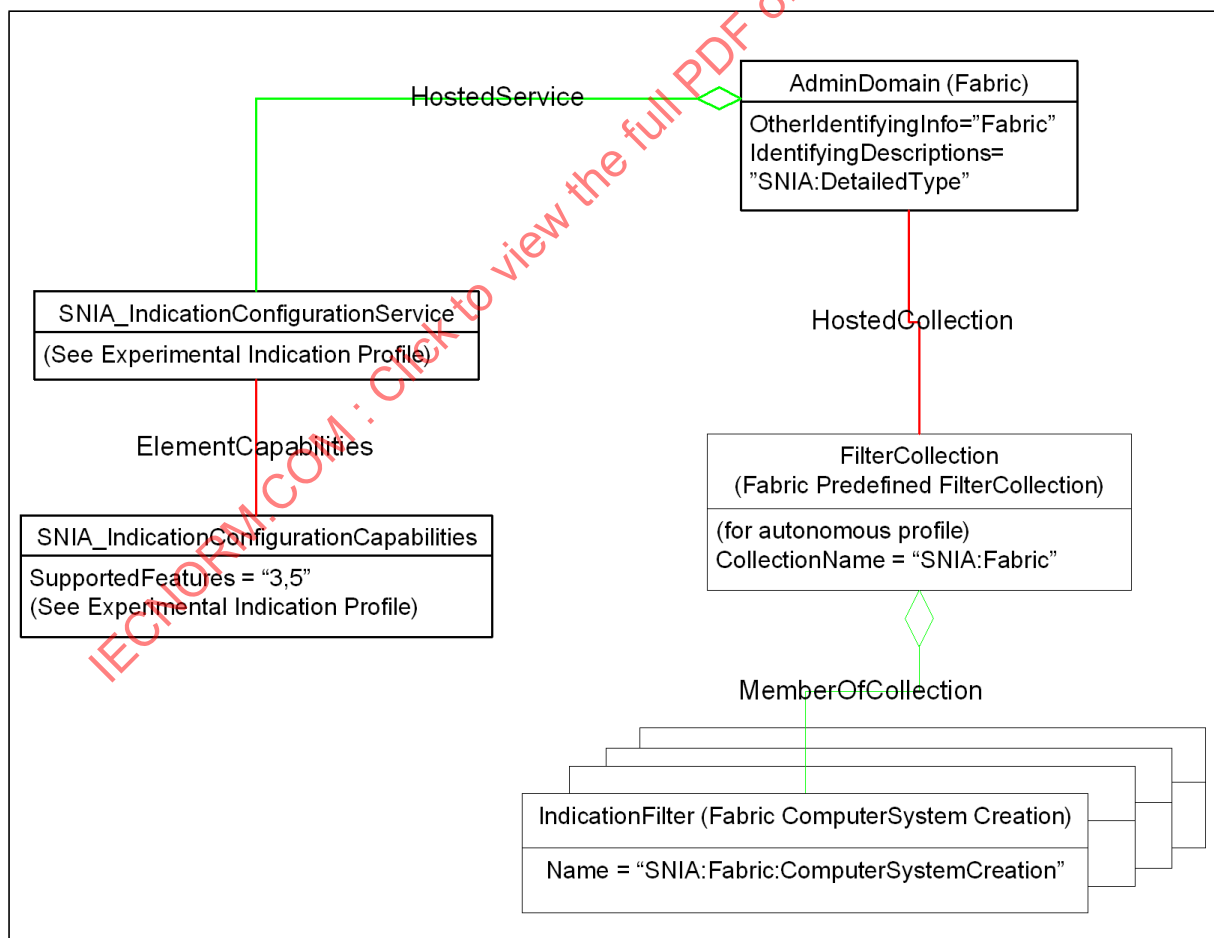


Figure 9 - Predefined Fabric IndicationFilters and Collection

The SNIA_IndicationConfigurationService is hosted on the fabric AdminDomain. The FilterCollection for fabric is also hosted on the fabric AdminDomain. The fabric FilterCollection has members which are all the predefined IndicationFilters supported by the implementation.

This shall include all Mandatory IndicationFilters of the Fabric Profile. But it should also include any IndicationFilter that has been predefined by the implementation. This may include conditional, optional or vendor specific IndicationFilters supported by the implementation.

The fabric FilterCollection shall have the CollectionName "SNIA:Fabric". Each of the predefined filters shall have the Name property as defined for the IndicationFilter. In Figure 9 the name of the IndicationFilter (Fabric ComputerSystem Creation) is "SNIA:Fabric:ComputerSystemCreation". For vendor specific IndicationFilters (not defined in this International Standard), the Name of the filter would be of the form ORG_ID:"PROFILE":UNIQUE_ID, where ORGID is the designation of the vendor that is providing the implementation.

EXPERIMENTAL

5.2 Health and Fault Management

The following classes report possible Health and Fault information through LifeCycle indications:

- ComputerSystem;
- FCPort.

These LifeCycle indications are more fully described in Table 4, "CIM Elements for Fabric".

Also in Table 4, "CIM Elements for Fabric" is a list of AlertIndications which may also be indicators for Health and Fault Management.

5.3 Cascading Considerations

None.

5.4 Supported Subprofiles and Package

Table 1 describes the supported profiles for Fabric.

Table 1 - Supported Profiles for Fabric

Profile Name	Organization	Version	Requirement	Description
Zone Control	SNIA	1.4.0	Optional	
Enhanced Zoning and Enhanced Zoning Control	SNIA	1.4.0	Optional	
FabricVirtualFabrics	SNIA	1.5.0	Optional	
N Port Virtualizer	SNIA	1.5.0	Optional	
FDMI	SNIA	1.4.0	Optional	
Fabric Path Performance	SNIA	1.2.0	Optional	
FabricSecurity	SNIA	1.2.0	Optional	
Indication	SNIA	1.5.0	Mandatory	
Experimental Indication	SNIA	1.5.0	Optional	Experimental.
Fabric Views	SNIA	1.5.0	Optional	Experimental.
Launch In Context	DMTF	1.0.0	Optional	Experimental. See DSP1102, version 1.0.0

5.5 Methods of this Profile

Not defined in this International Standard.

5.6 Client Considerations and Recipes

5.6.1 Fabric Identifier

The client needs to consider that the fabric identifier is not durable but is correlatable and may change over time. See *Information Technology – Storage Management, Part 2 Common Architecture* 8 Correlatable and Durable Names.

5.6.2 FCPort OperationalStatus

OperationalStatus is the property to indicate status and state for the FCPort (both Platform and Switch ports). The FCPort instance has one of the following Operational Statuses.

Table 2 - Port OperationalStatus

OperationalStatus	Description
OK	Port is online
Error	Port has a failure
Stopped	Port is disabled
InService	Port is in Self Test
Unknown	

5.6.3 Switch ComputerSystem OperationalStatus

OperationalStatus is the property to indicate status and state for switch ComputerSystems. The switch ComputerSystem instance has one of the following Operational Statuses and possibly one of the Subsidiary statuses, as shown in Table 3.

Table 3 - OperationalStatus for ComputerSystem

Operational Status	Possible Subsidiary Operational Status	Description
OK		The system has a good status.
OK	Stressed	The system is stressed, for example the temperature is over limit or there is too much IO in progress.
OK	Predictive Failure	The system will probably fail sometime soon.
Degraded		The system is operational but not at 100 % redundancy. A component has suffered a failure or something is running slow.
Error		An error has occurred causing the system to stop. This error may be recoverable with operator intervention.
Error	Non-recoverable error	A severe error has occurred. Operator intervention is unlikely to fix it.
Error	Supporting entity in error	A modeled element has failed.
InService		Switch is in Self Test.
No contact		The provider knows about the array but has not talked to it since last reboot.
Lost communication		The provider used to be able to communicate with the array, but has now lost contact.
Starting		The system is starting up.
Stopping		The system is shutting down.
Stopped		The data path is OK but shut down, the management channel is still working.

5.6.4 Discover The Fabric Topology

```

// This recipe describes how to build a topology graph of a fabric.
//
// 1. Identifies all the Switches and adds their objects paths and the
// object paths of the FC Ports belonging to these Switches to the $nodes
// array
//
// 2. Creates a suitable Association instance (e.g., a SystemDevice
// Association instance between a Switch and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array
//
// 3. Creates a map of all connected FC Ports (i.e., belonging to
// Switches
// that are ISL'd together and to Host HBAs and Storage System Front End
// Controllers)
//
// In this map, the FC Ports (i.e., the ones that are connected) are
// cross-connected.
//
// e.g., For a pair of FC Ports, one belonging to a Switch and the other
// belonging to a Host (HBA), the map indexed by the Switch Port WWN
// returns
// the Host (HBA) FC Port object path and the map indexed by the Host
// (HBA)
// FC Port WWN returns the Switch FC Port object path.
//
// Similar relationship exists between the pairs of FC Ports where one
// belongs to a Switch and the other belonging belongs to a Storage
// System
// Front End Controller and for FC Ports each of which belongs to a
// Switch.
//
// 4. Identifies all the Hosts and adds their objects paths to the $nodes
// array. Note that the object paths of the FC Ports (HBA Ports)
// belonging
// to these Hosts are already added to the $nodes array in step-3.
//
// 5. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Host and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the
// Association
// to the $links array.
//
// 6. Identifies all the Storage Systems and adds their objects paths to
// the
// $nodes array.
// Note that the object paths of the FC Ports (i.e., Front End Controller
// FC Ports) belonging to these Storage Systems are already added to the
// $nodes array in step-3.
//
// 7. Creates a suitable Association instance (e.g. a SystemDevice

```

```
// Association instance between a Storage System and a FC Port), setting
// its GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array.

// DESCRIPTION
// Create a map of how elements in a SAN are connected together via
// Fibre-ChannelFC ports.
//
// The map is built in array $attachedFcPorts->[], where the index is a
// WWN of any device port on the SAN, and the value at that index is
// the object path of the connected Switch or HBA or Storage System FC
// port.
//
// First find all the switches in a SAN. Get all the FC Ports for each
// switch and get the Attached FC Ports for each Switch FC Port. Save
// these
// device FC ports in the map described above.

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. All agents/namespaces supporting Fabric Profile previously
// identified
// using SLP. Do this for each CIMOM supporting Fabric Profile

switches[] = enumerateInstances("CIM_ComputerSystem", true, false, true,
                                true, null)

for #i in $switches[]
{
    if (!contains(5, $switches[#i].Dedicated))
        continue

    // only process switches, not other computer systems

    // Add the switch to the $nodes array

    $nodes.addIfNotAlreadyAdded ($switches[#i].getObjectPath());

    // Get all the SystemDevice associations between this switch and its
    // FC Ports

    $sysDevAssoc[] = ReferenceNames($switches[#i],
                                    "CIM_FCPort",
                                    "GroupComponent");

    // Add these associations to the $links array

    for #a in $sysDevAssoc->[]
    $links.addIfNotAlreadyAdded ($sysDevAssoc->[#a];

    $fcPorts->[] = AssociatorNames(
        $switches[#i].getObjectPath(),
        "CIM_SystemDevice",
        "CIM_FCPort",
```

```

    "GroupComponent",
    "PartComponent")
for #j in $fcPorts->[]
{

    // Add the FC Port in $nodes array

    $nodes.addIfNotAlreadyAdded (fcPorts->[#j];

    $protocolEndpoints->[] = AssociatorNames(
        fcPorts->[#j],
        "CIM_DeviceSAPIImplementation",
        "CIM_ProtocolEndpoint",
        "Antecedent",
        "Dependent");

    // NOTE - It is possible for this collection to be empty (i.e.,
        ports
    // that are not connected). It is possible for this collection to
    // have more than one element (loops attached to a switch port is
        the
    // most common example).

    if ($protocolEndpoints->[].length == 0)
        continue

    // Add the Protocol End Point to the nodes array.
    // Currently this recipe is designed to only save one
    // ProtocolEndpoint.

    $nodes.addIfNotAlreadyAdded (protocolEndpoints[0]);

    // Add the associations between the fcPort and the Protocol end
        point
    // to the links array

    $devSAPImplassoc[] = ReferenceNames($fcPorts->[#j],
        "CIM_ProtocolEndpoint",
        null);
    for #a in $devSAPImplassoc->[]
        $links.addIfNotAlreadyAdded ($devSAPImplassoc->[#a];

    $attachedProtocolEndpoints->[] = AssociatorNames(
        $protocolEndpoints->[0],
        "CIM_ActiveConnection",
        "CIM_ProtocolEndpoint",
        null, null)

    // Add the Attached Protocol End Point to the nodes array

    $nodes.addIfNotAlreadyAdded (attachedProtocolEndpoints->[0]);

```

```
// Add the associations between the Protocol end point and the
// Attached protocol endpoint to the links array

$actConnassoc[] = ReferenceNames($protocolEndpoint->[#0],
                                "CIM_ActiveConnection",
                                null);

for #a in $actConnassoc->[]
    $links.addIfNotAlreadyAdded ($actConnassoc->[#a];

// NOTE: role & resultRole are null as the direction of the
// association is not dictated by the specification

// $attachedFcPort is either a device FC port or an ISL'd switch
// FC
// port from another switch. We store this result is stored (i.e.
// which device FC Port is connected // to which switch FC Port)
// in
// a suitable data structure for subsequent correlation to ports
// discovered on devices.

for #k in $attachedProtocolEndpoints->[]
{
    $attachedFcPorts->[] = Associators(
        $attachedProtocolEndpoints->[#k],
        "CIM_DeviceSAPImplementation",
        "CIM_FCPort",
        "Dependent",
        "Antecedent",
        false,
        false,
        ["PermanentAddress"])
    $attachedFcPort = $attachedFcPorts[0] // Exactly one member
        guaranteed by model

    // Add the attached FC Port to the $nodes array
    if $attachedFcPort != null
        $nodes.addIfNotAlreadyAdded ($attachedFcPort);
}
}

Determine the active Zone Set in a SAN
// DESCRIPTION
// Traverse from the fabric to all zone sets, looking for
// the active zone set
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The fabric of interest (an AdminDomain) has been previously
// identified and defined in the $Fabric-> variable
```

```

$ZoneSets[] = Associators($Fabric->, "CIM_HostedCollection",
                        "CIM_ZoneSet", null, null, false, false, null)

for #i in $ZoneSets[] {
    if ($ZoneSet[#i].Active) {
        // <found active ZoneSet>
        // NOTE - there can be only one active ZoneSet in a fabric, though
        //           there may be none
        break
    }
}

```

5.7 Registered Name and Version

Fabric version 1.5.0 (Autonomous Profile).

5.8 CIM Elements

5.8.1 General

Table 4 describes the CIM elements for Fabric.

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
5.8.2 CIM_ActiveConnection	Mandatory	The association between ProtocolEndpoints representing the links between devices.
5.8.3 CIM_AdminDomain (Fabric)	Mandatory	AdminDomain representing the Fabric. Associated to RegisteredProfile.
5.8.4 CIM_AdminDomain (SAN)and	Mandatory	AdminDomain representing the SAN.
5.8.5 CIM_Component (Platform to Fabric)	Optional	Aggregates Hosts and Arrays (Platforms) in the AdminDomain that represents the Fabric.
5.8.6 CIM_Component (Switch to Fabric)	Mandatory	Aggregates Switches in the AdminDomain that represents the Fabric.
5.8.7 CIM_ComputerSystem (Host Platform)	Mandatory	The ComputerSystem representing the Host Platform.
5.8.8 CIM_ComputerSystem (Partitioned Switch)	Conditional	Conditional requirement: Required if the Virtual Fabrics profile is implemented. The ComputerSystem representing a Partitioned Switch.
5.8.9 CIM_ComputerSystem (Storage Platform)	Mandatory	The ComputerSystem representing the Storage Platform (e.g. an Array).
5.8.10 CIM_ComputerSystem (Switch)	Mandatory	The ComputerSystem representing the Switch.
5.8.11 CIM_ConnectivityCollection	Mandatory	Collects the ProtocolEndpoints of the fabric.
5.8.12 CIM_ContainedDomain	Mandatory	Associates a Fabric to a SAN.
5.8.13 CIM_DeviceSAPImplementation (Non-Switch to FCPort)	Mandatory	Associates the Non-Switch (Host or Storage) FCPort to the ProtocolEndpoint.
5.8.14 CIM_DeviceSAPImplementation (Switch to FCPort)	Mandatory	Associates the Switch FCPort to the ProtocolEndpoint.
5.8.15 CIM_ElementCapabilities (ZoneCapabilities to Fabric)	Mandatory	Associates ZoneCapabilities to a Fabric.

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
5.8.16 CIM_ElementCapabilities (ZoneCapabilities to Switch)	Mandatory	Associates ZoneCapabilities to a switch system.
5.8.17 CIM_ElementSettingData (ZoneMembershipSettingData to Zone)	Mandatory	Associates ZoneMembershipSettingData to the Zone.
5.8.18 CIM_FCActiveConnection	Optional	Experimental. The association between ProtocolEndpoints representing the links between fibre channel devices (including ISLs).
5.8.19 CIM_FCPort (Host FCPort)	Mandatory	Fibre Channel Port for Host Platforms.
5.8.20 CIM_FCPort (Host NPIV FCPort)	Optional	Experimental. A Host NPIV Fibre Channel Port for Host Platforms.
5.8.21 CIM_FCPort (Partitioned Switch FCPort)	Conditional	Conditional requirement: Support for the Switch Partitioning profile. Fibre Channel Port for Partitioned Switch.
5.8.22 CIM_FCPort (Storage FCPort)	Mandatory	Fibre Channel Port for Devices (Storage Platforms).
5.8.23 CIM_FCPort (Switch FCPort)	Mandatory	Fibre Channel Port for Switch.
5.8.24 CIM_FilterCollection (Fabric Predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This is a collection of predefined IndicationFilters to which a client may subscribe.
5.8.25 CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint)	Optional	Associates the ProtocolEndpoint to a Fabric AdminDomain.
5.8.26 CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)	Mandatory	Associates the ProtocolEndpoint to the hosting ComputerSystem.
5.8.27 CIM_HostedCollection (Fabric to ConnectivityCollection)	Mandatory	Associates the ConnectivityCollection to the AdminDomain representing the Fabric.
5.8.28 CIM_HostedCollection (Fabric to predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).
5.8.29 CIM_HostedCollection (System to LogicalPortGroup)	Mandatory	Associates the LogicalPortGroup to the ComputerSystem representing the platform (host or array) or the Fabric (if platforms are not implemented).
5.8.30 CIM_HostedCollection (Zones or ZoneSets to Fabric)	Mandatory	Associates the ZoneSets and Zones to the AdminDomain representing the Fabric.
5.8.31 CIM_HostedCollection (Zones or ZoneSets to Switch)	Mandatory	Associates the ZoneSets and Zones to the hosting System (the ComputerSystem representing the switch).
5.8.32 CIM_HostedDependency	Optional	Experimental. The association representing the relationship of the FCPort that hosts the FCPorts that are virtualized.

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
5.8.33 CIM_IndicationFilter (Fabric ComputerSystem Creation)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new computer system instance in the Fabric (Switch or Platform instances).
5.8.34 CIM_IndicationFilter (Fabric ComputerSystem Deletion)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a fabric computer system instance.
5.8.35 CIM_IndicationFilter (Fabric ComputerSystem OperationalStatus)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the modification of OperationalStatus of a computer system instance (switch or platform) in the Fabric.
5.8.36 CIM_IndicationFilter (Fabric FCPort Creation)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new FCPort instance in the Fabric.
5.8.37 CIM_IndicationFilter (Fabric FCPort Deletion)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a fabric FCPort instance.
5.8.38 CIM_IndicationFilter (Fabric FCPort OperationalStatus)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the modification of OperationalStatus of an FCPort instance in the fabric.
5.8.39 CIM_IndicationFilter (Fabric Fabric Added or Removed)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the addition or removal of a fabric.

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
5.8.40 CIM_IndicationFilter (Fabric Fabric Merge or Segmentation)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the merge of fabrics or segmentation of a fabric.
5.8.41 CIM_IndicationFilter (Fabric Switch Added or Removed)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the addition or removal of a switch to a fabric.
5.8.42 CIM_IndicationFilter (Fabric Switch Status Changed)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a switch in a fabric.
5.8.43 CIM_IndicationFilter (Fabric Zone Database Changed)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for zone database changes in a fabric.
5.8.44 CIM_IndicationFilter (Fabric ZoneSet Activated)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for activation of a ZoneSet in a fabric.
5.8.45 CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus)	Conditional	Deprecated. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a fabric (Switch or Platform) ComputerSystem instance.
5.8.46 CIM_IndicationFilter (WQL Fabric FCPort OperationalStatus)	Conditional	Deprecated. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a fabric FCPort instance.
5.8.47 CIM_LogicalPortGroup	Mandatory	Fibre Channel Node.

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
5.8.48 CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint)	Mandatory	Associates ConnectivityCollection to ProtocolEndpoint.
5.8.49 CIM_MemberOfCollection (LogicalPortGroup to FCPort)	Mandatory	Associates a LogicalPortGroup to its platform FC Ports.
5.8.50 CIM_MemberOfCollection (Predefined Filter Collection to Fabric Filters)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This associates the Fabric predefined FilterCollection to the predefined Filters supported by the Fabric.
5.8.51 CIM_MemberOfCollection (ZoneSet to Zone)	Mandatory	Associates a ZoneSet to its Zones.
5.8.52 CIM_ProtocolEndpoint	Mandatory	The endpoint of a link (ActiveConnection).
5.8.53 CIM_SystemDevice (Non-Switch FCPort to Fabric)	Optional	Associates non-Switch FCPorts to the System (Fabric).
5.8.54 CIM_SystemDevice (Non-Switch FCPort to Platform)	Optional	Associates non-Switch FCPorts to the ComputerSystem (Platform).
5.8.55 CIM_SystemDevice (Switch FCPort to Switch)	Mandatory	Associates Switch FCPorts to the ComputerSystem (Switch).
5.8.56 CIM_Zone (Active)	Mandatory	The active Zones being enforced by the Fabric.
5.8.57 CIM_Zone (Inactive)	Mandatory	The inactive Zones being enforced by the Fabric.
5.8.58 CIM_ZoneCapabilities	Mandatory	The Zoning Capabilities of the associated Fabric (or Switch).
5.8.59 CIM_ZoneMembershipSettingData	Mandatory	Defines the zone member.
5.8.60 CIM_ZoneSet (Active)	Mandatory	The active ZoneSets being enforced by the Fabric.
5.8.61 CIM_ZoneSet (Inactive)	Mandatory	The inactive ZoneSet.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	Creation of a ComputerSystem instance. See 5.8.33 CIM_IndicationFilter (Fabric ComputerSystem Creation).
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	Deletion of a ComputerSystem instance. See 5.8.34 CIM_IndicationFilter (Fabric ComputerSystem Deletion).
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort	Mandatory	Creation of a FC Port instance. See 5.8.36 CIM_IndicationFilter (Fabric FCPort Creation).
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort	Mandatory	Deletion of a FC Port instance. See 5.8.37 CIM_IndicationFilter (Fabric FCPort Deletion).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated WQL - Modification of OperationalStatus of a FC Port instance. See 5.8.46 CIM_IndicationFilter (WQL Fabric FCPort OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus	Mandatory	CQL - Modification of OperationalStatus of a FC Port instance. See 5.8.38 CIM_IndicationFilter (Fabric FCPort OperationalStatus).

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated WQL - Modification of OperationalStatus of a ComputerSystem instance. See 5.8.45 CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus	Mandatory	CQL - Modification of OperationalStatus of a ComputerSystem instance. See 5.8.35 CIM_IndicationFilter (Fabric ComputerSystem OperationalStatus).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC1'	Mandatory	CQL - Modification of Zone Database. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> 9.3.4.1, Message: Zone Database Changed. See also 5.8.43 CIM_IndicationFilter (Fabric Zone Database Changed).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC1'	Mandatory	Deprecated. Deprecated WQL - Modification of Zone Database. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> , 9.3.4.1 Message: Zone Database Changed.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC2'	Mandatory	CQL - ZoneSet Activated. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> , 9.3.4.2 Message: ZoneSet Activated. See 5.8.44 CIM_IndicationFilter (Fabric ZoneSet Activated).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC2'	Mandatory	Deprecated. Deprecated WQL - ZoneSet Activated. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> , 9.3.4.2 Message: ZoneSet Activated.
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC5'	Mandatory	Experimental. Switch Status Changed. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> , 9.3.4.5 Message: Switch Status Changed. See also 5.8.42 CIM_IndicationFilter (Fabric Switch Status Changed).

Table 4 - CIM Elements for Fabric

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC6'	Mandatory	Experimental. Fabric Merge/Segmentation. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> 9.3.4.6 Message: Fabric Merge/Segmentation See also 5.8.40 CIM_IndicationFilter (Fabric Fabric Merge or Segmentation).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC7'	Mandatory	Experimental. Switch Added/Removed. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> 9.3.4.7 Message: Switch Added/Removed See also 5.8.41 CIM_IndicationFilter (Fabric Switch Added or Removed).
SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC8';	Mandatory	Experimental. Fabric Added/Removed. See ISO/IEC 24775-2, <i>Information Technology – Storage Management, Part 2 Common Architecture</i> 9.3.4.8 Message: Fabric Added/Removed See also 5.8.39 CIM_IndicationFilter (Fabric Fabric Added or Removed).

5.8.2 CIM_ActiveConnection

The association between ProtocolEndpoints representing the links between devices (including ISLs). For loops, multiple ActiveConnections are instantiated as one to many relationships.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 5 describes class CIM_ActiveConnection.

Table 5 - SMI Referenced Properties/Methods for CIM_ActiveConnection

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The reference to the ProtocolEndpoint for one end of the link.
Dependent		Mandatory	The reference to the ProtocolEndpoint for the other end of the link.

5.8.3 CIM_AdminDomain (Fabric)

AdminDomain representing the fabric. This is a logical entity and can represent virtual fabrics.

Note that if the Virtual Fabrics Profile is implemented, OtherIdentifyInfo shall contain a Virtual Fabric ID with a 'SNIA:VF_ID' in the corresponding index of IdentifyingDescriptions.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Shall be associated to RegisteredProfile using ElementConformsToProfile association. The RegisteredProfile instance shall have RegisteredName set to 'Fabric', RegisteredOrganization set to 'SNIA', and RegisteredVersion set to '1.5.0'.

Table 6 describes class CIM_AdminDomain (Fabric).

Table 6 - SMI Referenced Properties/Methods for CIM_AdminDomain (Fabric)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name	C	Mandatory	WWN of Fabric.
NameFormat		Mandatory	Shall be 'WWN'.
ElementName		Optional	A user friendly name for the Fabric (implementation dependent).
OtherIdentifyingInfo		Mandatory	For a Fabric AdminDomain this property shall contain the value 'Fabric'. For Virtual Fabrics, one of the indices shall contain the value of the Virtual Fabric ID and in the corresponding index for OtherIdentifyingDescription the value 'SNIA:VF_ID'.
IdentifyingDescriptions		Mandatory	For a Fabric AdminDomain this property shall contain the value 'SNIA:DetailedType' in the index for the OtherIdentifyingInfo of 'Fabric'. For Virtual Fabrics, one of the indices shall contain the value 'SNIA:VF_ID' and in the corresponding index for OtherIdentifyingInfo the value of the Virtual Fabric ID.

5.8.4 CIM_AdminDomain (SAN)and

AdminDomain representing the SAN.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 7 describes class CIM_AdminDomain (SAN).

Table 7 - SMI Referenced Properties/Methods for CIM_AdminDomain (SAN)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name		Mandatory	An arbitrary name (implementation dependent).
NameFormat		Mandatory	Dependent on the arbitrary name chosen.
ElementName		Optional	A user friendly name for the SAN (implementation dependent).
OtherIdentifyingInfo		Mandatory	For a SAN AdminDomain this property shall contain the value 'SAN'.
IdentifyingDescriptions		Mandatory	For a SAN AdminDomain this property shall contain the value 'SNIA:DetailedType' in the index for the OtherIdentifyingInfo of 'SAN'.

5.8.5 CIM_Component (Platform to Fabric)

Aggregates Hosts and Arrays (Platforms) in the AdminDomain that represents the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 8 describes class CIM_Component (Platform to Fabric).

Table 8 - SMI Referenced Properties/Methods for CIM_Component (Platform to Fabric)

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	Reference to a Storage Platform or Host Platform ComputerSystem.
GroupComponent		Mandatory	Reference to the AdminDomain representing the Fabric (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').

5.8.6 CIM_Component (Switch to Fabric)

Aggregates Switches in the AdminDomain that represents the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 9 describes class CIM_Component (Switch to Fabric).

Table 9 - SMI Referenced Properties/Methods for CIM_Component (Switch to Fabric)

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	Reference to Switch (a ComputerSystem with Dedicated='5').
GroupComponent		Mandatory	Reference to the AdminDomain representing the Fabric (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').

5.8.7 CIM_ComputerSystem (Host Platform)

The ComputerSystem representing the Host Platform. This class is typically instantiated if the end device has populated the Fibre Channel Platform Database or FDMI.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 10 describes class CIM_ComputerSystem (Host Platform).

Table 10 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Host Platform)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name	C	Mandatory	The Platform Name or FDMI Host Name.
ElementName		Mandatory	The Platform Label.
NameFormat		Mandatory	
Dedicated		Mandatory	For a FC-GS-6 Platform Type of Host, this shall be 'Not Dedicated' (0) or 'Unknown' (1).

5.8.8 CIM_ComputerSystem (Partitioned Switch)

The ComputerSystem representing a Partitioned Switch. This is a constrained case of the Switch ComputerSystem. Instance of a Partitioned Switch shall have an entry of 'Virtual Switch' and an entry for the Virtual Fabric ID in the OtherIdentifyingInfo array property (as well as the DomainID for the Switch).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Virtual Fabrics profile is implemented.

Table 11 describes class CIM_ComputerSystem (Partitioned Switch).

Table 11 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned Switch)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	See the CreationClassName definition in 5.8.10 CIM_ComputerSystem (Switch).
Name	C	Mandatory	See the Name definition in 5.8.10 CIM_ComputerSystem (Switch).
ElementName		Mandatory	See the ElementName definition in 5.8.10 CIM_ComputerSystem (Switch).
NameFormat		Mandatory	See the NameFormat definition in 5.8.10 CIM_ComputerSystem (Switch).
OperationalStatus		Mandatory	See the OperationalStatus definition in 5.8.10 CIM_ComputerSystem (Switch).
OtherIdentifyingInfo		Mandatory	One element shall contain the DomainID stored in decimal format. Another value corresponding to 'SNIA:DetailedType' shall contain 'Virtual Switch'. Yet another element shall contain a Virtual Fabric ID corresponding to 'SNIA:VF_ID'. In addition, other 'SNIA:DetailedType' entries may also be in this property (e.g., 'Front Domain', 'Translate Domain' or 'Backbone').
Dedicated		Mandatory	See the Dedicated definition in 5.8.10 CIM_ComputerSystem (Switch).

Table 11 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned Switch)

Properties	Flags	Requirement	Description and Notes
IdentifyingDescriptions		Mandatory	One of the indices shall contain the value 'DomainID' and in the corresponding index for OtherIdentifyingInfo the value of the Domain ID. One of the other indices shall contain the value 'SNIA:DetailedType' and in the corresponding index for OtherIdentifyingInfo a value of 'Virtual Switch' for a Partitioned Switch. Yet another element shall contain 'SNIA:VF_ID' in the element that corresponds to the virtual fabric ID in OtherIdentifyingInfo.
EnabledState		Mandatory	Experimental. See the EnabledState definition in 13.8.3 CIM_ComputerSystem (Switch).
RequestedState		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the RequestedState definition in 5.8.3 CIM_AdminDomain (Fabric).
EnabledDefault		Optional	See the EnabledDefault definition in ...
RequestStateChange()		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the RequestStateChange definition in 13.8.3 CIM_ComputerSystem (Switch).

5.8.9 CIM_ComputerSystem (Storage Platform)

The ComputerSystem representing the Storage Platform (e.g. an Array). This class is typically instantiated if the end device has populated the Fibre Channel Platform Database.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 12 describes class CIM_ComputerSystem (Storage Platform).

Table 12 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Storage Platform)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name	C	Mandatory	The Platform Name.
ElementName		Mandatory	The Platform Label.
NameFormat		Mandatory	
Dedicated		Mandatory	For a FC-GS-6 Platform Type for storage subsystems, 'Storage' (3) or 'Unknown' (1).

5.8.10 CIM_ComputerSystem (Switch)

The ComputerSystem representing the Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 13 describes class CIM_ComputerSystem (Switch).

Table 13 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Switch)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name	C	Mandatory	The Switch WWN.
ElementName		Mandatory	The Switch Symbolic Name.
NameFormat		Mandatory	Shall be 'WWN'.
OperationalStatus		Mandatory	One of the defined values (2 3 6 8 9 10 11 12 13) shall be present in the array value.
OtherIdentifyingInfo		Mandatory	One element shall contain the DomainID stored in decimal format. Another value corresponding to 'SNIA:DetailedType' shall contain 'Front Domain', 'Translate Domain', 'Virtual Switch' or 'None'.
Dedicated		Mandatory	Shall be 5 (Switch).
IdentifyingDescriptions		Mandatory	One of the indices shall contain the value 'DomainID' and in the corresponding index for OtherIdentifyingInfo the value of the Domain ID. One of the other indices shall contain the value 'SNIA:DetailedType' and in the corresponding index for OtherIdentifyingInfo a value of 'Front Domain' for an Inter-Fabric Routing (IFR) Front Domain, 'Translate Domain' for an Inter-Fabric Routing (IFR) Translate Domain, 'Virtual Switch' for a Virtual Fabric Partitioned Switch, or 'None' when the other values don't apply.
EnabledState		Mandatory	Experimental. This property shall be 5 ('Not Applicable') when an implementation does not support the Switch Profile for this switch.
RequestedState		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the RequestedState definition in 13.8.3 CIM_ComputerSystem (Switch).
EnabledDefault		Optional	See the EnabledDefault definition in 13.8.3 CIM_ComputerSystem (Switch).
RequestStateChange()		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the RequestStateChange definition in 13.8.3 CIM_ComputerSystem (Switch).

5.8.11 CIM_ConnectivityCollection

Collects the ProtocolEndpoints of the fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 14 describes class CIM_ConnectivityCollection.

Table 14 - SMI Referenced Properties/Methods for CIM_ConnectivityCollection

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Optional	Not required, can be the Fabric WWN.

5.8.12 CIM_ContainedDomain

Associates one or more Fabrics to a SAN.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 15 describes class CIM_ContainedDomain.

Table 15 - SMI Referenced Properties/Methods for CIM_ContainedDomain

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	The reference to the AdminDomain representing the Fabric (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').
GroupComponent		Mandatory	The reference to the AdminDomain representing the SAN (OtherIdentifyingInfo contains 'SAN' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').

5.8.13 CIM_DeviceSAPImplementation (Non-Switch to FCPort)

Associates the Non-Switch (Host or Storage) FCPort to the ProtocolEndpoint.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 16 describes class CIM_DeviceSAPImplementation (Non-Switch to FCPort).

Table 16 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (Non-Switch to FCPort)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the ProtocolEndpoint.
Antecedent		Mandatory	Reference to the Non-Switch FCPort (Host or Storage).

5.8.14 CIM_DeviceSAPImplementation (Switch to FCPort)

Associates the Switch FCPort to the ProtocolEndpoint.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 17 describes class CIM_DeviceSAPImplementation (Switch to FCPort).

Table 17 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation (Switch to FCPort)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the ProtocolEndpoint.
Antecedent		Mandatory	Reference to the Switch FCPort or Virtual Switch FCPort.

5.8.15 CIM_ElementCapabilities (ZoneCapabilities to Fabric)

Associates the ZoneCapabilities to a System. The system is the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 18 describes class CIM_ElementCapabilities (ZoneCapabilities to Fabric.).

Table 18 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ZoneCapabilities to Fabric)

Properties	Flags	Requirement	Description and Notes
Capabilities		Mandatory	Reference to ZoneCapabilities.
ManagedElement		Mandatory	Reference to a Fabric AdminDomain (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').

5.8.16 CIM_ElementCapabilities (ZoneCapabilities to Switch)

Associates the ZoneCapabilities to a System. The system normally is the AdminDomain representing the Fabric, but in some cases where the Zone Database is not a fabric entity, it maybe hosted on a ComputerSystem representing the Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 19 describes class CIM_ElementCapabilities (ZoneCapabilities to Switch).

Table 19 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (ZoneCapabilities to Switch)

Properties	Flags	Requirement	Description and Notes
Capabilities		Mandatory	Reference to ZoneCapabilities.
ManagedElement		Mandatory	Reference to a Switch ComputerSystem.

5.8.17 CIM_ElementSettingData (ZoneMembershipSettingData to Zone)

Associates ZoneMembershipSettingData to the Zone.

Created By: Extrinsic: AddZoneMemberSettingData

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 20 describes class CIM_ElementSettingData (ZoneMembershipSettingData to Zone).

Table 20 - SMI Referenced Properties/Methods for CIM_ElementSettingData (ZoneMembershipSettingData to Zone)

Properties	Flags	Requirement	Description and Notes
SettingData		Mandatory	Reference to a ZoneMembershipSettingData.
ManagedElement		Mandatory	Reference to the Zone.

5.8.18 CIM_FCActiveConnection

Experimental. The association between ProtocolEndpoints representing the links between fibre channel devices (including ISLs). For loops and NPIV, multiple ActiveConnections are instantiated as one to many relationships.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 21 describes class CIM_FCActiveConnection.

Table 21 - SMI Referenced Properties/Methods for CIM_FCActiveConnection

Properties	Flags	Requirement	Description and Notes
Discriminator		Optional	An array property enumeration used to discriminate the context in which the ActiveConnection is instantiated. Values supported by this International Standard include 0 (Unknown), 2 (None), 3 (NPIV), 4 (Chassis), 5 (Virtual Fabric) or 6 (IFR).
Antecedent		Mandatory	The reference to the ProtocolEndpoint for one end of the link.
Dependent		Mandatory	The reference to the ProtocolEndpoint for the other end of the link.

5.8.19 CIM_FCPort (Host FCPort)

Fibre Channel Port for Host Platforms.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 22 describes class CIM_FCPort (Host FCPort).

Table 22 - SMI Referenced Properties/Methods for CIM_FCPort (Host FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.
ElementName		Optional	Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.
PermanentAddress	CD	Mandatory	Fibre Channel Port WWN. Expressed as 16 un-separated upper case hex digits (see Table 4 for more information about formats).
NetworkAddresses	C	Mandatory	Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 4 for more information about formats).
OperationalStatus		Mandatory	One of the defined values (0 2 6 10 11) shall be present in the array value.
PortType		Mandatory	The specific port type (0 1 10 11 12 13 14 15 16 17 18) currently enabled (from FC-GS-6 Port.Type).
LinkTechnology		Mandatory	Shall be 4 ('FC').
SupportedFC4Types		Optional	An array of integers indicating the Fibre Channel FC-4 protocols supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.] EXPERIMENTAL .
SupportedCOS		Optional	An array of integers indicating the Fibre Channel Classes of Service that are supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '2' or '3' for FC-SB-2 Channel or FC-SB-2 Control Unit ports, respectively.] EXPERIMENTAL .

5.8.20 CIM_FCPort (Host NPIV FCPort)

Experimental. A Host NPIV Fibre Channel Port for Host Platforms.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 23 describes class CIM_FCPort (Host NPIV FCPort).

Table 23 - SMI Referenced Properties/Methods for CIM_FCPort (Host NPIV FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.
ElementName		Optional	Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.
PermanentAddress	CD	Mandatory	Fibre Channel Port WWN. Expressed as 16 un-separated upper case hex digits (see Table 4 for more information about formats).
NetworkAddresses	C	Mandatory	Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 4 for more information about formats).
OperationalStatus		Mandatory	One of the defined values (0 2 6 10 11) shall be present in the array value.
PortType		Mandatory	The specific port type (0 1 10 11 12 13 14 15 16 17 18) currently enabled (from FC-GS-6 Port.Type).
LinkTechnology		Mandatory	Shall be 4 ('FC').
PortDiscriminator		Mandatory	Experimental. For NPIV Host FCPorts this shall contain '7' (NPIV).
SupportedFC4Types		Optional	An array of integers indicating the Fibre Channel FC-4 protocols supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.] EXPERIMENTAL.
SupportedCOS		Optional	An array of integers indicating the Fibre Channel Classes of Service that are supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '2' or '3' for FC-SB-2 Channel or FC-SB-2 Control Unit ports, respectively.] EXPERIMENTAL.

5.8.21 CIM_FCPort (Partitioned Switch FCPort)

Fibre Channel Port for Partitioned Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Support for the Switch Partitioning profile.

Table 24 describes class CIM_FCPort (Partitioned Switch FCPort).

**Table 24 - SMI Referenced Properties/Methods for CIM_FCPort
(Partitioned Switch FCPort)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in 5.8.23 CIM_FCPort (Switch FCPort).
SystemName		Mandatory	See the SystemName definition in 5.8.23 CIM_FCPort (Switch FCPort).
CreationClassName		Mandatory	See the CreationClassName definition in 5.8.23 CIM_FCPort (Switch FCPort).
DeviceID		Mandatory	See the DeviceID definition in 5.8.23 CIM_FCPort (Switch FCPort).
ElementName		Mandatory	See the ElementName definition in 5.8.23 CIM_FCPort (Switch FCPort).
PermanentAddress	CD	Mandatory	See the PermanentAddress definition in 5.8.23 CIM_FCPort (Switch FCPort).
Speed		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the Speed definition in 13.8.12 CIM_FCPort (Switch FCPort).
OperationalStatus		Mandatory	See the OperationalStatus definition in 5.8.23 CIM_FCPort (Switch FCPort).
PortType		Mandatory	See the PortType definition in 5.8.23 CIM_FCPort (Switch FCPort).
LinkTechnology		Mandatory	See the LinkTechnology definition in 5.8.23 CIM_FCPort (Switch FCPort).
EnabledState		Mandatory	Experimental. See the EnabledState definition in 5.8.23 CIM_FCPort (Switch FCPort).
DetailedPortState		Conditional	Experimental. Conditional requirement: Required if the Switch profile is implemented. See Table 130 - DetailedPortState for FCPort.
PortAvailability		Optional	Experimental. See Table 131 - PortAvailability for FCPort.
RequestedState		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the RequestedState definition in 13.8.12 CIM_FCPort (Switch FCPort).
EnabledDefault		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the EnabledDefault definition in 13.8.12 CIM_FCPort (Switch FCPort).
MaxSpeed		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the MaxSpeed definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortNumber		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the PortNumber definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortDiscriminator		Mandatory	Experimental. This array property identifies the context in which this FCPort is instantiated. For a Partitioned Switch FCPort, the values may be '3' (VF), '6' (IFR (Virtual)) or '7' (NPV).
RequestStateChange()		Conditional	Conditional requirement: Support for the Switch Profile and a non-null value in FCPortCapabilities.RequestedStatesSupported. See the RequestStateChange definition in 13.8.12 CIM_FCPort (Switch FCPort).

5.8.22 CIM_FCPort (Storage FCPort)

Fibre Channel Port for Storage Platforms.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 25 describes class CIM_FCPort (Storage FCPort).

Table 25 - SMI Referenced Properties/Methods for CIM_FCPort (Storage FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.
ElementName		Optional	Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.
PermanentAddress	CD	Mandatory	Fibre Channel Port WWN. Expressed as 16 un-separated upper case hex digits (see Table 4 for more information about formats).
NetworkAddresses	C	Mandatory	Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 4 for more information about formats).
OperationalStatus		Mandatory	One of the defined values (0 2 6 10 11) shall be present in the array value.
PortType		Mandatory	The specific port type (0 1 10 11 12 13 14 15 16 17 18) currently enabled (from FC-GS-6 Port.Type).
LinkTechnology		Mandatory	Shall be 4 ('FC').
SupportedFC4Types		Optional	An array of integers indicating the Fibre Channel FC-4 protocols supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.] EXPERIMENTAL .
SupportedCOS		Optional	An array of integers indicating the Fibre Channel Classes of Service that are supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '2' or '3' for FC-SB-2 Channel or FC-SB-2 Control Unit ports, respectively.] EXPERIMENTAL .

5.8.23 CIM_FCPort (Switch FCPort)

Fibre Channel Port for Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 26 describes class CIM_FCPort (Switch FCPort).

Table 26 - SMI Referenced Properties/Methods for CIM_FCPort (Switch FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory.	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.
ElementName		Mandatory	Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.
PermanentAddress	CD	Mandatory	Fibre Channel Port WWN.
Speed		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the Speed definition in 13.8.12 CIM_FCPort (Switch FCPort).
OperationalStatus		Mandatory	One of the defined values (0 2 6 10 11) shall be present in the array value.
PortType		Mandatory	The specific port type (0 1 10 11 12 13 14 15 16 17 18) currently enabled (from FC-GS-6 Port.Type).
LinkTechnology		Mandatory	This shall be 4 ('FC').
EnabledState		Mandatory	Experimental. This property shall be 5 ('Not Applicable') when an implementation does not support the Switch Profile for the switch this port is on.
DetailedPortState		Conditional	Experimental. Conditional requirement: Required if the Switch profile is implemented. See Table 130 - DetailedPortState for FCPort.
PortAvailability		Optional	Experimental. See Table 131 - PortAvailability for FCPort.
RequestedState		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the RequestedState definition in 13.8.12 CIM_FCPort (Switch FCPort).
EnabledDefault		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the EnabledDefault definition in 13.8.12 CIM_FCPort (Switch FCPort).
MaxSpeed		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the MaxSpeed definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortNumber		Conditional	Conditional requirement: Required if the Switch profile is implemented. See the PortNumber definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortDiscriminator		Optional	Experimental. This array property identifies the context in which this FCPort is instantiated. For this standard, the values may be 0 (Unknown), 2 (Not applicable), 3 (VF), 4 (FCIP), 5 (IFR), 6 (IFR (Virtual)), 7 (NPV), 8 (Internal) or 9 (Chassis).
RequestStateChange()		Conditional	Conditional requirement: Support for the Switch Profile and a non-null value in FCPortCapabilities.RequestedStatesSupported. See the RequestStateChange definition in 13.8.12 CIM_FCPort (Switch FCPort).

5.8.24 CIM_FilterCollection (Fabric Predefined FilterCollection)

Experimental. This is a collection of predefined IndicationFilters to which a client may subscribe. A Fabric implementation shall indicate support for predefined FilterCollections by the SNIA_IndicationConfigurationCapabilities.FeaturesSupported = '5' (Predefined Filter Collections).

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 27 describes class CIM_FilterCollection (Fabric Predefined FilterCollection).

Table 27 - SMI Referenced Properties/Methods for CIM_FilterCollection (Fabric Predefined FilterCollection)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Shall specify the unique identifier for an instance of this class within the Implementation namespace.
CollectionName		Mandatory	The value of CollectionName shall be 'SNIA:Fabric'.

5.8.25 CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint)

Associates the ProtocolEndpoint to the Fabric AdminDomain for those systems not registered in the Platform Database or discovered through FDMI.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 28 describes class CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint).

Table 28 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (AdminDomain to ProtocolEndpoint)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the ProtocolEndpoint.
Antecedent		Mandatory	Reference to the Fabric AdminDomain (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions).

5.8.26 CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)

Associates the ProtocolEndpoint to the hosting System. The hosting System is a ComputerSystem for the Switch or Platform.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 29 describes class CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint).

**Table 29 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint
(ComputerSystem to ProtocolEndpoint)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the ProtocolEndpoint.
Antecedent		Mandatory	Reference to the Switch, Storage Platform or Host Platform ComputerSystem.

5.8.27 CIM_HostedCollection (Fabric to ConnectivityCollection)

Associates the ConnectivityCollection to the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 30 describes class CIM_HostedCollection (Fabric to ConnectivityCollection).

**Table 30 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Fabric to ConnectivityCollection)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the ConnectivityCollection.
Antecedent		Mandatory	Reference to the Fabric AdminDomain (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').

5.8.28 CIM_HostedCollection (Fabric to predefined FilterCollection)

Experimental.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 31 describes class CIM_HostedCollection (Fabric to predefined FilterCollection).

**Table 31 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Fabric to predefined FilterCollection)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the predefined FilterCollection for the Fabric.
Antecedent		Mandatory	Reference to the Fabric AdminDomain.

5.8.29 CIM_HostedCollection (System to LogicalPortGroup)

Associates the LogicalPortGroup to the ComputerSystem representing the platform (host or array) or the Fabric (if platforms are not implemented).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 32 describes class CIM_HostedCollection (System to LogicalPortGroup).

**Table 32 - SMI Referenced Properties/Methods for CIM_HostedCollection
(System to LogicalPortGroup)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the LogicalPortGroup.
Antecedent		Mandatory	Reference to the Platform ComputerSystem or Fabric AdminDomain.

5.8.30 CIM_HostedCollection (Zones or ZoneSets to Fabric)

Associates the ZoneSets and Zones to the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 33 describes class CIM_HostedCollection (Zones or ZoneSets to Fabric).

**Table 33 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Zones or ZoneSets to Fabric)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the Zone or ZoneSet.
Antecedent		Mandatory	Reference to the Fabric AdminDomain (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').

5.8.31 CIM_HostedCollection (Zones or ZoneSets to Switch)

Associates the ZoneSets and Zones to the hosting System (the ComputerSystem representing the switch).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 34 describes class CIM_HostedCollection (Zones or ZoneSets to Switch).

**Table 34 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Zones or ZoneSets to Switch)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the Zone or ZoneSet.
Antecedent		Mandatory	Reference to the Switch ComputerSystem.

5.8.32 CIM_HostedDependency

Experimental. The association representing the relationship of the FCPort that hosts the FCPorts that are virtualized. This association is required for all FCPorts that are instantiated as part of NPIV.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 35 describes class CIM_HostedDependency.

Table 35 - SMI Referenced Properties/Methods for CIM_HostedDependency

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	A reference to the hosting Host FCPort (physical fibre channel port).
Dependent		Mandatory	A reference to the hosted Host NPIV FCPort (virtualized fibre channel port with PortDiscriminator=7).

5.8.33 CIM_IndicationFilter (Fabric ComputerSystem Creation)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new fabric computer system instance. This would represent the addition of a switch or platform computer system to the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 36 describes class CIM_IndicationFilter (Fabric ComputerSystem Creation).

Table 36 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric ComputerSystem Creation)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> , 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:ComputerSystemCreation'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 36 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric ComputerSystem Creation)**

Properties	Flags	Requirement	Description and Notes
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.34 CIM_IndicationFilter (Fabric ComputerSystem Deletion)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a fabric computer system instance. This would represent the removal of a switch or platform computer system from the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 37 describes class CIM_IndicationFilter (Fabric ComputerSystem Deletion).

**Table 37 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric ComputerSystem Deletion)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:ComputerSystemDeletion'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 37 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric ComputerSystem Deletion)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in section <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.35 CIM_IndicationFilter (Fabric ComputerSystem OperationalStatus)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a fabric ComputerSystem instance. This would represent the status change to a switch or platform computer system in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 38 describes class CIM_IndicationFilter (Fabric ComputerSystem OperationalStatus).

**Table 38 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric ComputerSystem OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:ComputerSystemOperationalStatus'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 38 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric ComputerSystem OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.36 CIM_IndicationFilter (Fabric FCPort Creation)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new fabric FCPort instance. This would represent the addition of a switch or platform FCPort to the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 39 describes class CIM_IndicationFilter (Fabric FCPort Creation).

**Table 39 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric FCPort Creation)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:FCPortCreation'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 39 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric FCPort Creation)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.37 CIM_IndicationFilter (Fabric FCPort Deletion)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a fabric FCPort instance. This would represent the removal of a switch or platform FCPort from the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 40 describes class CIM_IndicationFilter (Fabric FCPort Deletion).

**Table 40 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric FCPort Deletion)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:FCPortDeletion'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

Table 40 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric FCPort Deletion)

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.38 CIM_IndicationFilter (Fabric FCPort OperationalStatus)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a fabric FCPort instance. This would represent the status change to a switch or platform FCPort in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 41 describes class CIM_IndicationFilter (Fabric FCPort OperationalStatus).

Table 41 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric FCPort OperationalStatus)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:FCPortOperationalStatus'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 41 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric FCPort OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.39 CIM_IndicationFilter (Fabric Fabric Added or Removed)

Experimental. This is the WQL 'pre-defined' CIM_IndicationFilter instance for addition or removal of a fabric. This would represent the addition or removal of a fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 42 describes class CIM_IndicationFilter (Fabric Fabric Added or Removed).

**Table 42 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric Fabric Added or Removed)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:FabricAddedOrRemoved'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 42 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric Fabric Added or Removed)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC8'.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.40 CIM_IndicationFilter (Fabric Fabric Merge or Segmentation)

Experimental. This is the WQL 'pre-defined' CIM_IndicationFilter instance for the merge of fabrics or segmentation of a fabric. This would represent the merge of fabrics or segmentation of a fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 43 describes class CIM_IndicationFilter (Fabric Fabric Merge or Segmentation).

**Table 43 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric Fabric Merge or Segmentation)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:FabricMergeOrSegmentation'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC6'.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.41 CIM_IndicationFilter (Fabric Switch Added or Removed)

Experimental. This is the WQL 'pre-defined' CIM_IndicationFilter instance for the addition or removal of a switch to a fabric. This would represent the addition or removal of a switch to a fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 44 describes class CIM_IndicationFilter (Fabric Switch Added or Removed).

Table 44 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Switch Added or Removed)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:SwitchAddedOrRemoved'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC7'.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.42 CIM_IndicationFilter (Fabric Switch Status Changed)

Experimental. This is the WQL 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a switch in a fabric. This would represent the change of OperationalStatus of a switch in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 45 describes class CIM_IndicationFilter (Fabric Switch Status Changed).

Table 45 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Fabric Switch Status Changed)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:SwitchStatusChanged'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC5'.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.43 CIM_IndicationFilter (Fabric Zone Database Changed)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for zone database changes in a fabric. This would represent changes to zones, zonesets and zone aliases in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 46 describes class CIM_IndicationFilter (Fabric Zone Database Changed).

**Table 46 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric Zone Database Changed)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:ZoneDatabaseChange'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in section <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC1'.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.44 CIM_IndicationFilter (Fabric ZoneSet Activated)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for activation of a ZoneSet in a fabric. This would represent activation of a ZoneSet in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 47 describes class CIM_IndicationFilter (Fabric ZoneSet Activated).

**Table 47 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Fabric ZoneSet Activated)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:ZoneSetActivated'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_AlertIndication WHERE OwningEntity='SNIA' and MessageID='FC2'.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.45 CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus)

Deprecated. This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a fabric (Switch or Platform) computer system instance. This would represent the OperationalStatus changes of a switch or platform computer system in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 48 describes class CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus).

Table 48 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Fabric ComputerSystem OperationalStatus)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:ComputerSystemOperationalStatus WQL'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus.
QueryLanguage		Mandatory	This shall be 'WQL' or 'SMI-S V1.0'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.46 CIM_IndicationFilter (WQL Fabric FCPort OperationalStatus)

Deprecated. This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a fabric FCPort instance. This would represent the OperationalStatus changes of a switch or platform FCPort in the fabric. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 49 describes class CIM_IndicationFilter (WQL Fabric FCPort OperationalStatus).

**Table 49 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(WQL Fabric FCPort OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Fabric:FCPortOperationalStatusWQL'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus.
QueryLanguage		Mandatory	This shall be 'WQL' or 'SMI-S V1.0'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3 <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

5.8.47 CIM_LogicalPortGroup

Represents the Fibre Channel Node. Associated to the host system by the HostedCollection Association. The hosting System is either a ComputerSystem representing the Platform or the AdminDomain representing the fabric in the case for those systems not registered in the Platform Database or discovered through FDMI (but available through the Name Server/ Management Server).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 50 describes class CIM_LogicalPortGroup.

Table 50 - SMI Referenced Properties/Methods for CIM_LogicalPortGroup

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
Name	CD	Mandatory	Fibre Channel Node WWN.
NameFormat		Mandatory	Shall be 'WWN'.
ElementName	N	Mandatory	Node Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.

5.8.48 CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 51 describes class CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint).

Table 51 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (ConnectivityCollection to ProtocolEndpoint)

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the ConnectivityCollection.
Member		Mandatory	Reference to the ProtocolEndpoint.

5.8.49 CIM_MemberOfCollection (LogicalPortGroup to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 52 describes class CIM_MemberOfCollection (LogicalPortGroup to FCPort).

Table 52 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (LogicalPortGroup to FCPort)

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the LogicalPortGroup.
Member		Mandatory	Reference to the Non-Switch (Host or Storage) FCPort.

5.8.50 CIM_MemberOfCollection (Predefined Filter Collection to Fabric Filters)

Experimental. This associates the Fabric predefined FilterCollection to the predefined Filters supported by the Fabric.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 53 describes class CIM_MemberOfCollection (Predefined Filter Collection to Fabric Filters).

Table 53 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Fabric Filters)

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the Fabric predefined FilterCollection.
Member		Mandatory	Reference to the predefined IndicationFilters of the Fabric.

5.8.51 CIM_MemberOfCollection (ZoneSet to Zone)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 54 describes class CIM_MemberOfCollection (ZoneSet to Zone).

Table 54 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (ZoneSet to Zone)

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the ZoneSet.
Member		Mandatory	Reference to the Zone.

5.8.52 CIM_ProtocolEndpoint

The endpoint of a link (ActiveConnection). ProtocolEndpoint shall be implemented when an ActiveConnection exists. It may be implemented if no ActiveConnections exist.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 55 describes class CIM_ProtocolEndpoint.

Table 55 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.

Table 55 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint

Properties	Flags	Requirement	Description and Notes
Name	CD	Mandatory	The Fibre Channel Port WWN.
NameFormat		Mandatory	'WWN'.
ProtocolIFType		Mandatory	Shall be 56 (Fibre channel).

5.8.53 CIM_SystemDevice (Non-Switch FCPort to Fabric)

Associates non-Switch FCPorts to the System (Fabric).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 56 describes class CIM_SystemDevice (Non-Switch FCPort to Fabric).

Table 56 - SMI Referenced Properties/Methods for CIM_SystemDevice (Non-Switch FCPort to Fabric)

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	The reference to a Host or Storage FCPort.
GroupComponent		Mandatory	The reference to the Fabric (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions).

5.8.54 CIM_SystemDevice (Non-Switch FCPort to Platform)

Associates non-switch FCPorts to the Platform ComputerSystem.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 57 describes class CIM_SystemDevice (Non-Switch FCPort to Platform).

Table 57 - SMI Referenced Properties/Methods for CIM_SystemDevice (Non-Switch FCPort to Platform)

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	The reference to a Host or Storage FcPort.
GroupComponent		Mandatory	The reference to the Platform (Storage or Host) System.

5.8.55 CIM_SystemDevice (Switch FCPort to Switch)

Associates the Switch FCPort to the ComputerSystem (Switch).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 58 describes class CIM_SystemDevice (Switch FCPort to Switch).

Table 58 - SMI Referenced Properties/Methods for CIM_SystemDevice (Switch FCPort to Switch)

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	The reference to the switch FCPort.
GroupComponent		Mandatory	The reference to the Switch ComputerSystem.

5.8.56 CIM_Zone (Active)

The active Zones being enforced by the Fabric.

Created By: Extrinsic: ActivateZoneSet

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 59 describes class CIM_Zone (Active).

Table 59 - SMI Referenced Properties/Methods for CIM_Zone (Active)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	The Zone Name.
ZoneType		Mandatory	The Zone Type.
Active		Mandatory	Shall be TRUE. Indicates that this ZoneSet is active.

5.8.57 CIM_Zone (Inactive)

The inactive Zones being enforced by the Fabric.

Created By: Extrinsic: CreateZone

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 60 describes class CIM_Zone (Inactive).

Table 60 - SMI Referenced Properties/Methods for CIM_Zone (Inactive)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	The Zone Name.
ZoneType		Mandatory	The Zone Type.
Active		Mandatory	Shall be FALSE. Indicates that this ZoneSet is inactive.

5.8.58 CIM_ZoneCapabilities

The Zoning Capabilities of the associated Fabric (or Switch).

ZoneCapabilities exposes the capabilities of the AdminDomain representing the Fabric for active zoning and the capabilities of the ComputerSystem representing the Switch or AdminDomain representing the Fabric for Zone Set Database.

If a ZoneCapability property is not applicable or does not explicitly exists (e.g. the capability is limited only by a memory size), the property is NULL.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 61 describes class CIM_ZoneCapabilities.

Table 61 - SMI Referenced Properties/Methods for CIM_ZoneCapabilities

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
MaxNumZoneSets		Optional	The maximum number of ZoneSets in the Zone Set Database. NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.
MaxNumZone		Optional	The maximum number of Zones in the Zone Set Database. NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.
MaxNumZoneMembers		Optional	The maximum number of ZoneMembers in the Zone Set Database. All ZoneMembers included in both Zones and ZoneAliases are counted, while the same ZoneMember included in multiple Zones or ZoneAliases is counted only once. NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.
MaxNumZoneAliases		Optional	The maximum number of ZoneAliases in the Zone Set Database NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.
ZoneNameMaxLen		Mandatory	The maximum length for the name of a ZoneAlias (NamedAddressCollection.ElementName), Zone (Zone.ElementName) or ZoneSet (ZoneSet.ElementName) the Fabric (or Switch) are capable of supporting.

Table 61 - SMI Referenced Properties/Methods for CIM_ZoneCapabilities

Properties	Flags	Requirement	Description and Notes
ZoneNameFormat		Mandatory	The name format of a ZoneAlias *NamedAddressCollection.ElementName), Zone (Zone.ElementName) or ZoneSet (ZoneSet.ElementName) supported by either the Fabric (or the Switch).
MaxNumZonesPerZoneSet		Optional	The maximum number of Zones per ZoneSet. NULL should be returned in such cases when the property is not applicable or the number is not limited explicitly.
SupportedConnectivityMemberTypes		Mandatory	An array containing the supported connectivity member types supported which include Permanent Address (WWN), Switch Port ID (Domain:Port in base10), Network Address (FCID), Logical Port Group (Node WWN).

5.8.59 CIM_ZoneMembershipSettingData

Defines the zone member.

Created By: Extrinsic: AddZoneMemberSettingData

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 62 describes class CIM_ZoneMembershipSettingData.

Table 62 - SMI Referenced Properties/Methods for CIM_ZoneMembershipSettingData

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ConnectivityMemberType		Mandatory	Permanent Address (WWN), Switch Port ID (Domain:Port in base10), Network Address (FCID).
ConnectivityMemberID	C	Mandatory	The value of the WWN, Domain/Port, or FCID.

5.8.60 CIM_ZoneSet (Active)

The active ZoneSet being enforced by the Fabric.

Created By: Extrinsic: ActivateZoneSet

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 63 describes class CIM_ZoneSet (Active).

Table 63 - SMI Referenced Properties/Methods for CIM_ZoneSet (Active)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	The ZoneSet name.
Active		Mandatory	Shall be TRUE. Indicates that this ZoneSet is active and members cannot be changed.

5.8.61 CIM_ZoneSet (Inactive)

The inactive ZoneSets.

Created By: Extrinsicpre-definedsic: CreateZoneSet

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 64 describes class CIM_ZoneSet (Inactive).

Table 64 - SMI Referenced Properties/Methods for CIM_ZoneSet (Inactive)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	The ZoneSet name.
Active		Mandatory	Shall be FALSE. Indicates that this ZoneSet is inactive.

STABLE

STABLE
6 Enhanced Zoning and Enhanced Zone Control Subprofile**6.1 Description**

This profile describes the additional zoning functions for enhanced zoning. Note that Sessions are normally part of enhanced zoning, but are included in the base fabric profile to address the various types of zoning operations into a single object model. In this subprofile, then only Zone Alias is added.

6.2 Health and Fault Management

None.

6.3 Cascading Considerations

None.

6.4 Dependencies on Profiles, Subprofiles, and Packages

Table 65 describes the supported profiles for Enhanced Zoning and Enhanced Zoning Control.

Table 65 - Supported Profiles for Enhanced Zoning and Enhanced Zoning Control

Profile Name	Organization	Version	Requirement	Description
Zone Control	SNIA	1.4.0	Mandatory	

Support for Clause 7 Zone Control Subprofile is mandatory for the Enhanced Zoning and Enhanced Zoning Control Subprofile.

6.5 Methods of this Profile**6.5.1 CreateZoneAlias**

The method creates a ZoneAlias and the association HostedCollection. The newly created association, HostedCollection, associates the ZoneAlias to the same AdminDomain the ZoneService is hosted to. For the newly created ZoneAlias, the Active property is always set to false.

```
CreateZoneAlias(
    [IN] string CollectionAlias,
    [OUT] CIM_NamedAddressCollection ref ZoneAlias);
```

6.5.2 AddZoneAlias

Adds to the Zone the specified ZoneAlias.

```
AddZoneAlias(
    [IN] CIM_Zone ref Zone,
    [IN] CIM_NamedAddressCollection ref ZoneAlias);
```

6.6 Client Considerations and Recipes**6.6.1 Create a ZoneAlias**

```
// DESCRIPTION
// Create zone alias and add new zone member based on
// the parameters collected by the CIM Client.
// Before any operations can be imposed on the zoning
// service, a session is requested and obtained from the zone
```

```
// service. Create a new ZoneAlias. The session may not be ended if
// the ZoneAlias is empty, so add a zone member to the new ZoneAlias.
// The session is released when the operations are
// completed.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1.The system of interest, either the fabric (AdminDomain)
// or the switch (ComputerSystem), has been
// previously identified and defined in the
// $System-> variable
// 2.The name of the new zone alias is defined in the
// #ZoneAliasName variable
// 3. The zone member type is defined in the #ConnectivityMemberType
// variable
// 4. The zone member Id of the new zone member is defined in the
// #ConnectiivtyMemberID variable

// 1. Get the ZoneService and start a session
$ZoneServices->[] = AssociatorNames(
    $System->,
    "CIM_HostedService",
    "CIM_ZoneService", null, null)

// Assumption 1 above guarantees there is a zone service for this
// system. the fabric and switch profiles that there is no more than
// one ZoneService for this system
$ZoneService-> = $ZoneServices[0]

if(!&startSession($ZoneService->))
{
    return
}

// 2. Create the ZoneAlias
%InArguments["CollectionAlias"] = #ZoneAliasName
#status = InvokeMethod(
    $ZoneService->,
    "CreateZoneAlias",
    %InArguments[],
    %OutArguments[])

$ZoneAlias-> = %OutArguments["ZoneAlias"]
if(#status != 0)
    // ERROR!

// 3. Create or locate a ZoneMembershipSettingData
%InArguments["ConnectivityMemberType"] = #ConnectivityMemberType
%InArguments["ConnectivityMemberID"] = #ConnectivityMemberID
%InArguments["SystemSpecificCollection"] = $ZoneAlias->
```

```

#status = InvokeMethod($ZoneService->,
    "CreateZoneMembershipSettingData",
    %InArguments[], %OutArguments[])

// 4. Add to zone alias if not created as a member of the zone alias
//     Zone member reference is set accordingly in the output arguments.

$ZoneMember-> = %OutArguments["ZoneMembershipSettingData"]

if (#status != 0)
    // ERROR!

// 5. End the session gracefully
&endSession($ZoneService->)
// 6. Verify that the ZoneAlias exists in the database
try{
    GetInstance($ZoneAlias->)
} catch (CIM_ERR_NOT_FOUND) {
    // error
}

```

6.6.2 Delete a ZoneAlias

```

// DESCRIPTION
// Delete a zone alias.
// Before any operations can be imposed on the zoning service, a
// session is requested and obtained from the zone service.
// The session is released when the operations are completed.
//
// if the deletion fails, it may be because the Zone Alias is not empty.
// In this case, remove all members from the alias by deleting the
// ElementSettingData associations, and try the deletion again.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1.The system of interest, either the fabric (AdminDomain)
//    or the switch (ComputerSystem), has been
//    previously identified and defined in the
//    $System-> variable
// 2.The object name of the zone alias to be deleted is
//    defined in the $ZoneAlias-> variable

// 1. Get the zone service and start a session
$ZoneServices->[] = AssociatorNames(
    $System->,
    "CIM_HostedService",
    "CIM_ZoneService",
    null,
    null)

// Assumption 1 above guarantees there is a zone service for this
// system. the fabric and switch profiles that there is no more than

```

```
// one ZoneService for this system
$ZoneService-> = $ZoneServices[0]

if(!&startSession($ZoneService->))
{
    return
}

// 2. Attempt to delete the alias
try{
    DeleteInstance($ZoneAlias->)
}catch(CIM_ERR_FAILED){
    // Try to remove any zone members in the alias
    // via the ElementSettingData association
    $ZoneMembers->[] = referenceNames($ZoneAlias->,
        "CIM_ElementSettingData",
        null)
    for #j in $ZoneMembers->[] {
        DeleteInstance($ZoneMembers[#j])
    }
    // Try again
    DeleteInstance($ZoneAlias->)
}

// 3. End Session
&endSession($ZoneService->)
// verify that the deletion occurred
try{
    GetInstance($ZoneAlias->)
}catch(CIM_ERR_NOT_FOUND){
    //expect exception
    return
}
// error!!
```

6.7 Registered Name and Version

Enhanced Zoning and Enhanced Zoning Control version 1.4.0 (Component Profile).

6.8 CIM Elements

6.8.1 General

Table 66 describes the CIM elements for Enhanced Zoning and Enhanced Zoning Control.

Table 66 - CIM Elements for Enhanced Zoning and Enhanced Zoning Control

Element Name	Requirement	Description
6.8.2 CIM_ElementSettingData (ZoneMembershipSettingData to NamedAddressCollection)	Mandatory	Associates ZoneMembershipSettingData to the NamedAddressCollection representing the ZoneAlias.
6.8.3 CIM_HostedCollection (AdminDomain to Collection)	Mandatory	Associates the NameAddressCollection representing the Zone Alias to the AdminDomain.
6.8.4 CIM_HostedCollection (ComputerSystem to Collection)	Mandatory	Associates the NameAddressCollection representing the Zone Alias to the System.
6.8.5 CIM_MemberOfCollection	Mandatory	Associates NamedAddressCollection with Zone.
6.8.6 CIM_NamedAddressCollection	Mandatory	The Zone Alias.
6.8.7 CIM_ZoneService (Zone Service)	Mandatory	The service that allows for all of the zoning configuration changes.

6.8.2 CIM_ElementSettingData (ZoneMembershipSettingData to NamedAddressCollection)

Associates ZoneMembershipSettingData to the NamedAddressCollection representing the ZoneAlias.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 67 describes class CIM_ElementSettingData (ZoneMembershipSettingData to NamedAddressCollection).

Table 67 - SMI Referenced Properties/Methods for CIM_ElementSettingData (ZoneMembershipSettingData to NamedAddressCollection)

Properties	Flags	Requirement	Description and Notes
SettingData		Mandatory	
ManagedElement		Mandatory	

6.8.3 CIM_HostedCollection (AdminDomain to Collection)

Associates the NamedAddressCollection representing the Zone Alias to the AdminDomain representing the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 68 describes class CIM_HostedCollection (AdminDomain to Collection).

**Table 68 - SMI Referenced Properties/Methods for CIM_HostedCollection
(AdminDomain to Collection)**

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	Reference to the Fabric AdminDomain (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').
Dependent		Mandatory	Reference to the NamedAddressCollection representing the Zone Alias.

6.8.4 CIM_HostedCollection (ComputerSystem to Collection)

Associates the NamedAddressCollection representing the Zone Alias to the ComputerSystem representing the switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 69 describes class CIM_HostedCollection (ComputerSystem to Collection).

**Table 69 - SMI Referenced Properties/Methods for CIM_HostedCollection
(ComputerSystem to Collection)**

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	A reference to a Switch ComputerSystem.
Dependent		Mandatory	Reference to the NamedAddressCollection representing the Zone Alias.

6.8.5 CIM_MemberOfCollection

Associates NamedAddressCollection with Zone.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 70 describes class CIM_MemberOfCollection.

Table 70 - SMI Referenced Properties/Methods for CIM_MemberOfCollection

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	
Member		Mandatory	

6.8.6 CIM_NamedAddressCollection

The Zone Alias.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 71 describes class CIM_NamedAddressCollection.

Table 71 - SMI Referenced Properties/Methods for CIM_NamedAddressCollection

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
CollectionAlias		Mandatory	The Zone Alias Name.

6.8.7 CIM_ZoneService (Zone Service)

The service that allows for all of the zoning configuration changes. The definition in this profile adds additional methods to the same service defined in Zone Control.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 72 describes class CIM_ZoneService (Zone Service).

Table 72 - SMI Referenced Properties/Methods for CIM_ZoneService (Zone Service)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	The Class Name.
Name		Mandatory	Opaque.
CreateZoneAlias()		Mandatory	
AddZoneAlias()		Mandatory	

STABLE

STABLE

7 Zone Control Subprofile

7.1 Description

This profile includes extrinsic methods for creating Zone Sets, Zones, and Zone Members and adding Zones to Zone Sets and Zone Members to Zones. Additionally SMI-S defines intrinsic methods for the removing of Zone Members from Zones and Zone Aliases, Zones from Zone Sets, and deleting Zone Members, Zones, and Zone Sets. The profile also includes methods to allow a client to request or release a lock on the fabric for zoning configuration changes.

When an Inactive ZoneSet is "Activated", new instances representing the Active Zone Set and Active Zones are generated from the Inactive Zone Set definition (where a switch may prune the referenced Zone Set collapsing aliases, removes empty zones, etc.).

When a new Zone Set is "Activated", the instances representing the previous active Zone Set no longer exists.

This profile also includes the property ConfigurationID on the ZoneService. The ConfigurationID is used to identify a unique point in time identifier of the zoning configuration. It may be a timestamp, a generation identifier, or a checksum. The ConfigurationID shall change whenever the zoning configuration changes. The value of the property is used as an input to ActivateZoneSet() and SessionControl() to allow the client to confirm that the change requested is consistent with the last discovery the client made of the zoning configuration.

In the case where the Inactive Zone Sets are hosted on a switch, the client cannot know which Inactive Zone Set was used to define the current Active Zone Set. Also if two Inactive Zone Sets with the same name are hosted on two different switches, the definitions maybe completely different.

7.2 Durable Names and Correlatable IDs of the Profile

None.

7.3 Instrumentation Requirements

The agent shall support the use case defined in 7.8.

7.4 Health and Fault Management

None.

7.5 Cascading Considerations

None.

7.6 Dependencies on Profiles, Subprofiles, and Packages

Related Profiles for Zone Control: Not defined in this International Standard.

7.7 Methods of this Profile

7.7.1 CreateZoneSet

The method creates a ZoneSet and associates it to the System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.

```
CreateZoneSet (  
    [IN] string ZoneSetName,  
    [OUT] CIM_ZoneSet ref ZoneSet);
```

7.7.2 CreateZone

The method creates a Zone and associates it to System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.

```
CreateZone (
    [IN] string ZoneName,
    [IN] uint16 ZoneType,
    [IN] uint16 ZoneSubType,
    [OUT] CIM_Zone ref Zone);
```

7.7.3 CreateZoneMembershipSettingData

The method creates a ZoneMembershipSettingData (a zone member) and adds it to the specified Zone or NamedAddressCollection representing a Zone Alias. The ConnectivityMemberID is dependent upon the ConnectivityMemberType.

For Fibre Channel, the ConnectivityMemberType of "PermanentAddress", the ConnectivityMemberID is the NxPort WWN; for ConnectivityMemberType of "NetworkAddress", the ConnectivityMemberID is the NXPort Address ID; for ConnectivityMemberType of "SwitchPortID", the ConnectivityMemberID is "Domain:PortNumber".

```
CreateZoneMembershipSettingData (
    [IN] uint16 ConnectivityMemberType,
    [IN] string ConnectivityMemberID,
    [IN] CIM_SystemSpecificCollection ref
        SystemSpecificCollection,
    [OUT] CIM_ZoneMembershipSettingData ref
        ZoneMembershipSettingData);
```

7.7.4 AddZone

The method adds to the specified ZoneSet the specified Zone. Adding a Zone to a ZoneSet, extends the zone enforcement definition of the ZoneSet to include the members of that Zone. If adding the Zone is successful, the Zone should be associated to the ZoneSet by MemberOfCollection.

```
AddZone (
    [IN] CIM_ZoneSet ref ZoneSet,
    [IN] CIM_Zone ref Zone);
```

7.7.5 AddZoneMembershipSettingData

The method adds to the specified Zone or NamedAddressCollection representing the Zone Alias the specified ZoneMembershipSettingData (a zone member).

```
AddZoneMembershipSettingData (
    [IN] CIM_SystemSpecificCollection ref
        SystemSpecificCollection,
    [IN] CIM_ZoneMembershipSettingData ref
        ZoneMembershipSettingData);
```

EXPERIMENTAL

7.7.6 ActivateZoneSetWithJob

This method activates the specified ZoneSet with the expectation that a job will be created. It has an optional input, ConfigurationID, which is used to confirm that the ZoneSet being

activated matches the definition the application previously discovered or changed. If the ConfigurationID passed in matches the ConfigurationID associated to the current zone definitions, then the fabric attempts to activate the ZoneSet. If the ConfigurationIDs do not match, the method exits without activating the ZoneSet. Once a ZoneSet is activated, a ZoneSet with the property Active set to true, its associated Zones with the property Active set to true, and the Zone's associated ZoneMembershipSettingData are instantiated.

This method may run asynchronously due to the length of the operation. This typically occurs when the session is ended and the changes are committed. When the method is run asynchronously, the value "Method parameters checked - job started" is returned and the output parameter Job is populated with a reference to the job instance.

ActivateZoneSet shall be supported outside of a session. ActivateZoneSet being called within a session is implementation specific.

Calling ActivateZoneSet outside of a session while a session is open is implementation specific.

```
uint32 ActivateZoneSetWithJob (
    [IN] CIM_ZoneSet ref ZoneSet,
    [IN] boolean Activate,
    [IN,OUT] string ConfigurationID,
    [OUT] CIM_ConcreteJob REF Job,
    [IN] datetime TimeoutPeriod)
```

EXPERIMENTAL

7.7.7 ActivateZoneSet

This method activates the specified ZoneSet without a Job. ActivateZoneSetWithJob replaces this method. It is supported for backwards compatibility and should not be used in future implementations.

Once a ZoneSet is activated, a ZoneSet with the property Active set to true, its associated Zones with the property Active set to true, and the Zone's associated ZoneMembershipSettingData are instantiated.

ActivateZoneSet shall be supported outside of a session. ActivateZoneSet being called within a session is implementation specific.

Calling ActivateZoneSet outside of a session while a session is open is implementation specific.

```
uint32 ActivateZoneSet (
    [IN] CIM_ZoneSet ref ZoneSet,
    [IN] boolean Activate )
```

EXPERIMENTAL

7.7.8 SessionControlWithJob

This method allows a client to request or release a lock on the fabric for zoning configuration changes. The method has an optional input, ConfigurationID, which is used to confirm that the ZoneSet being activated matches the definition the application previously discovered or changed. If the ConfigurationID passed in matches the ConfigurationID associated to the current zone definitions, then the fabric attempts to grant the lock. If the ConfigurationIDs do not match, the method exits without attempting to have the fabric grant the lock.

As described in FC-GS-6, in the context of Enhanced Zoning Management, management actions to a Zone Server (e.g., write access to the Zoning Database) shall occur only inside a GS session. Clients executing zoning management operations shall use fabric sessions

cooperatively if the Information Technology – Storage Management agent supports it. (If the value of SessionState is 4 (“Not Applicable”) then no cooperative session usage is possible).

Before a client executes zoning management operations (intrinsic or extrinsic methods), the client shall request a new session and wait for the request to be granted. To request a new session, first wait until the property “SessionState” of the fabric’s ZoneService is 3 (“Ended”) and the property “RequestedSessionState” is 5 “No Change”. Then call SessionControl with RequestedSessionState = 2 (“Started”). Once zoning management operations are completed, the client shall release the session to enable the provider to propagate changes to the fabric, and to allow other clients to perform management operations. To end a session and commit the changes, call SessionControl with RequestedSessionState = 3 (“Ended”). To abort a sequence of zoning management operations without updating the fabric, call SessionControl with RequestedSessionState = 4 (“Terminated”).

SMI-S agents shall block on calls to SessionControl until the request is fulfilled. For example, an error may occur while committing changes to a fabric, i.e., after a call to SessionControl with RequestedSessionState = 3 (“Ended”). The method cannot return until the session has ended, so that a CIM error can be returned if a problem occurs. While the method is in progress, another client may read the value of the RequestedSessionState property and see the value set by the method currently in progress. Once the request is fulfilled, the RequestedSessionState property is set to value 5 “No Change”, regardless of the value in the setInstance operation.

Sessions can timeout. The session timeout behavior and settings are defined by FC-SW-5 in the section discussing mapping GS sessions for Enhanced Zoning Management.

This method may run asynchronously due to the length of the operation. This shall occur when the session is “Ended” to commit the changes. It shall not be used for any other case. When the method is run asynchronously, the value “Method parameters checked - job started” is returned and the output parameter Job is populated with a reference to the job instance.

A SMI-S agent may raise an error if these client cooperation rules are not followed. For the purposes of a SMI-S agent, a series of requests from the same authenticated entity are considered to be from a single client. An agent may verify that such a series corresponds to the sequence described above and raise the error CIM_ERR_FAILED at any time if the sequence is violated.

```
uint32 SessionControlWithJob (
    [IN,
        ValueMap {"2", "3", "4"},
        Values {"Started", "Ended", "Terminated"}]
    uint16 RequestedSessionState,
    [IN,OUT] string ConfigurationID,
    [OUT] CIM_ConcreteJob REF Job,
    [IN] datetime TimeoutPeriod)
```

EXPERIMENTAL

7.7.9 SessionControl

The method enables a client to request a lock of the fabric to begin zoning configuration changes. It is supported for legacy implementations. SessionControlWithJob replaces this method. It is supported for backwards compatibility only and should not be used in future implementations.

This method allows a client to request or release a lock on the fabric for zoning configuration changes. As described in FC-GS-6, in the context of Enhanced Zoning Management, management actions to a Zone Server (e.g., write access to the Zoning Database) shall occur only inside a GS session. Clients executing zoning management operations shall use fabric sessions cooperatively if the Information Technology – Storage Management agent supports it.

(If the value of SessionState is 4 ("Not Applicable") then no cooperative session usage is possible).

Before a client executes zoning management operations (intrinsic or extrinsic methods), the client shall request a new session and wait for the request to be granted. To request a new session, first wait until the property "SessionState" of the fabric's ZoneService is 3 ("Ended") and the property "RequestedSessionState" is 5 "No Change". Then call SessionControl with RequestedSessionState = 2 ("Started"). Once zoning management operations are completed, the client shall release the session to enable the provider to propagate changes to the fabric, and to allow other clients to perform management operations. To end a session and commit the changes, call SessionControl with RequestedSessionState = 3 ("Ended"). To abort a sequence of zoning management operations without updating the fabric, call SessionControl with RequestedSessionState = 4 ("Terminated").

SMI-S agents shall block on calls to SessionControl until the request is fulfilled. For example, an error may occur while committing changes to a fabric, i.e., after a call to SessionControl with RequestedSessionState = 3 ("Ended"). The method cannot return until the session has ended, so that a CIM error can be returned if a problem occurs. While the method is in progress, another client may read the value of the RequestedSessionState property and see the value set by the method currently in progress. Once the request is fulfilled, the RequestedSessionState property is set to value 5 "No Change", regardless of the value in the setInstance operation.

Sessions can timeout. The session timeout behavior and settings are defined by INCITS FC-SW-5 in the section discussing mapping GS sessions for Enhanced Zoning Management.

An SMI-S agent may raise an error if these client cooperation rules are not followed. For the purposes of an SMI-S agent, a series of requests from the same authenticated entity are considered to be from a single client. An agent may verify that such a series corresponds to the sequence described above and raise the error CIM_ERR_FAILED at any time if the sequence is violated.

```
uint32 SessionControl (
    [IN,
    ValueMap {"2", "3", "4"},
    Values {"Started", "Ended", "Terminated"}]
    uint16 RequestedSessionState);;
```

7.7.10 Intrinsics for removing a zone from a zone set

As shown in Figure 7 in the Fabric Profile, a zone is a member of a zone set if there is a "CIM_MemberOfCollection" association from the zone set to the zone. To remove a zone from a zone set, delete the instance of the association "CIM_MemberOfCollection" using the intrinsic operation deleteInstance.

7.7.11 Intrinsics for removing a zone alias from a zone

A zone alias is a member of a zone if there is a "CIM_MemberOfCollection" association from the zone to the zone alias. To remove a zone alias from a zone set, delete the instance of the association "CIM_MemberOfCollection" using the intrinsic operation deleteInstance.

7.7.12 Intrinsics for removing a zone member from a zone or zone alias

Zone members are represented by CIM_ZoneMembershipSettingData instances. No instance of CIM_ZoneMembershipSettingData exists unless it is associated to a zone or zone alias by a CIM_ElementSettingData association. However, an instance of CIM_ZoneMembershipSettingData may be associated to more than one zone or zone alias.

Removing a zone member from a zone or zone alias is equivalent to deleting the instance of the CIM_ElementSettingData association. Delete the instance using the intrinsic operation deleteInstance.

If this is the last instance of a CIM_ElementSettingData association for a particular CIM_ZoneMembershipSettingData, do not delete the instance of CIM_ZoneMembershipSettingData. It is the provider's responsibility to clean up these structures.

7.7.13 Intrinsic for deleting a zone member

Zone members are represented by CIM_ZoneMembershipSettingData instances associated to zones or zone aliases via CIM_ElementSettingData associations. To delete a zone member (and remove it from any zones or zone aliases from which it is a member) use the CIM operation deleteInstance to delete the instance of CIM_ZoneMembershipSettingData.

Do not delete the corresponding instances of the CIM_ElementSettingData. It is the provider's responsibility to clean up these structures.

7.7.14 Intrinsic for deleting a zone, zone alias, or zone set

Use the intrinsic operation deleteInstance to delete a zone, zone alias or zone set. Clients are allowed to delete zones or zone aliases that are members of collections (zones or zone sets). Clients are allowed to delete the last member of a zone or zone set, leaving the collection empty.

A zone set or zone cannot be deleted if it is currently active (the error would be CIM_ERR_FAILED). Some implementations may prohibit deleting zonesets, zones or zone aliases that still have members (the error would be CIM_ERR_FAILED). When a zone, zone alias or zone set is deleted, the client does not have to delete the corresponding instances of CIM_MemberOfCollection or CIM_HostedCollection. It is the provider's responsibility to clean up these structures.

7.8 Client Considerations and Recipes

7.8.1 General

Many agent implementations do not allow Zone, a ZoneAlias or a Zone Set to be defined empty. Since the methods defined in SMI-S do not support creating a Zone Set with a Zone and a Zone with a Zone Member, the SessionControl method should be used to build a Zone Definition that is inter-operable. This is done by calling ZoneSession() to "Start" defining or updating the Zone Definition. The client then calls the appropriate methods as necessary to build the desired Zone Definition. For example, calling CreateZoneSet() to create a new Zone Set, CreateZone() to create a new Zone, AddZoneToZoneSet() to add the newly created Zone to the newly created Zone Set, and CreateZoneMembershipSettingData() to create and add a new Zone Member to the newly created Zone. Upon completion of the new zoning definition, ZoneControl is called again to "End" the session. The changes to the Zone Definition would then be applied to the Zone Set Database. This set of calls would create a Zone Definition where the Zone and ZoneSet are not empty and would be interoperable across all agent implementations.

7.8.2 Create or delete zones Common Functions

```
// DESCRIPTION
//
// Common functions used by the recipes below.
//
// startSession: attempt to start fabric session if required;
// returns false if attempt fails; returns true if attempt succeeds
// or if session control is unnecessary
//
```

```
// endSession: finalize fabric session if required; returns false
//   if attempt fails; returns true if attempt succeeds or if session
//   control is unnecessary
//
//
// findActiveZoneSet: routine to find the active zoneset
// on a fabric, and return the reference to it
//

// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// None

sub boolean startSession ($ZoneService->)
{
    $ZoneService = GetInstance($ZoneService->, false, false, false, null)

    // session statuses
    #Ended = 3
    #NotApplicable = 4

    // requested session statuses
    #Started = 2
    #NoChange = 5

    if ($ZoneService.SessionState == #NotApplicable)
        return true // no session control implemented by this agent

    if ($ZoneService.SessionState != #Ended)
        return false // fabric session is in use by another client or
            agent

    if ($ZoneService.RequestedSessionState != #NoChange)
        return false // another client has already requested session

    %InArguments["RequestedSessionState"] = #Started

    #status = InvokeMethod($ZoneService->, "SessionControl",
        %InArguments, %OutArguments)
    if (#status != 0) // e.g. "Failed"
        return false

    $ZoneService = GetInstance($ZoneService->, false, false, false, null)
    if ($ZoneService.SessionState != #Started)
        return false

    return true
}
```

```

// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// None

sub boolean endSession ($ZoneService->) {
    $ZoneService = GetInstance($ZoneService->, false, false, false, null)

    // session statuses
    #Started = 2
    #NotApplicable = 4

    // requested session statuses
    #End = 3

    if ($ZoneService.SessionStatus == #NotApplicable){
        return true // no need for session control

    if ($ZoneService.SessionStatus != #Started)
        return false // no session started by this client

    %InArguments["RequestedSessionState"] = #End
    #status = InvokeMethod($ZoneService, "SessionControl", %InArguments,
                           %OutArguments)
    if (#status != 0) // e.g. "Failed"
        return false

    // Do not wait, or even check, for SessionState to have value "Ended"
    // as
    // a) InvokeMethod will block till done (or failed) anyway
    // b) Before the check can be made, session may already be started
    // by another client

    return true
}

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The reference to the fabric on which the active
// zoneset it to be sought is already known in
// the input variable $Fabric. Calling code
// should verify that the returned reference is non-null
//
sub Ref findActiveZoneSet($Fabric->){
    $ActiveZoneSet->=null
    $ZoneSets[] = Associators(
        $Fabric->,
        "CIM_HostedCollection",
        "CIM_ZoneSet",
        null,
        null,
        false,

```

```
// there may be no active zoneset
if(0 < ZoneSets[].size()){
    for(#i in $ZoneSets[]){
        if(true==$ZoneSets[#i].Active){
            $ActiveZoneSet->=nameof($ZoneSets[#i])
            break
        }
    }
}

return $ActiveZoneSet->
```

[illegible]

```

// Assumption 1 (above) guarantees there is a zone service for this
// System, Fabric Profile mandates there is no more than one zone
// service for this System
$ZoneService-> = $ZoneService->[0]

// Start the session
if (!&startSession($ZoneService->)) {
    <ERROR! Failed to start zone session>
}

// 2. Create a ZoneMembershipSettingData
%InArguments["ConnectivityMemberType"] = #ConnectivityMemberType
%InArguments["ConnectivityMemberID"] = #ConnectivityMemberID
%InArguments["SystemSpecificCollection"] = $Zone->
#status = InvokeMethod($ZoneService->,
    "CreateZoneMembershipSettingData",
    %InArguments[], %OutArguments[])

if (#status != 0){
    <ERROR! call to method CreateZoneMembershipSettingData failed
    #status>
}

// 3. Store the returned object path for verification
$ZoneMember-> = %OutArguments["ZoneMembershipSettingData"]

// 4. End session successfully
if (!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}

// 5. Verify that the zonemember exist within the specified zone

$ZoneMembers->[]=associatorNames(
    $Zone->,
    "CIM_ElementSettingData",
    "CIM_ZoneMembershipSettingData",
    "ManagedElement",
    "SettingData" )
if(!contains($ZoneMember->,$ZoneMembers[])){
    <ERROR! Failed to verify zone member created>
}

```

7.8.4 Create new Zone, add new Zone Member, and add to existing ZoneSet

```

// DESCRIPTION
// Create new Zone, add new Zone Member, and add to existing ZoneSet
//
// Assume the client has already invoked some logic to determine which
// System (fabric or switch) will host the zone database and zone
// service to be used. Request and obtain a fabric session from the

```

```
// zone service. Create a new Zone using an extrinsic method. The
// session may not be ended if any zone is empty, so add a zone member
// to the new zone. The session also may not be ended unless every
// zone is a member of at least one zone set, so add the new zone to
// an existing zone set. Then close the fabric session.
//
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The System hosting the zone database (ComputerSystem or
// AdminDomain) has been previously identified and defined in the
// $System-> variable
//
// 2. The name for a new zone is defined in the #ZoneName variable
//
// 3. The type for the new zone is defined in the #ZoneType variable
//
// 4. The sub type for the new zone is defined in the #ZoneSubType
// variable
//
// 5. The zone member type is defined in the #ConnectivityMemberType
// variable
//
// 6. The zone member id of the new zone member is defined in the
// #ConnectivityMemberID variable
//
// 7. An existing zoneSet is defined in the $ZoneSet-> variable
//
// FUNCTIONS

// 1. Get the Zone Service and start the session
$ZoneServices->[] = AssociatorNames($System->, "CIM_HostedService",
                                     "CIM_ZoneService", null, null)

// Assumption 1 (above) guarantees there is a zone service for this
// System, Fabric Profile mandates there is no more than one zone
// service for this System
$ZoneService-> = $ZoneServices->[0]

    if (!&startSession($ZoneService->)) {
        <ERROR! Failed to start zone session>
    }

// 2. Create a zone
%InArguments["ZoneName"] = #ZoneName
%InArguments["ZoneType"] = #ZoneType
%InArguments["ZoneSubType"] = #ZoneSubType
InvokeMethod($ZoneService->, "CreateZone", %InArguments[],
              %OutArguments[])
$Zone-> = $OutArguments["Zone"]
```

```

// 3. Create a ZoneMembershipSettingData
%InArguments["ConnectivityMemberType"] = #ConnectivityMemberType
%InArguments["ConnectivityMemberID"] = #ConnectivityMemberID
%InArguments["SystemSpecificCollection"] = $Zone->
#status = InvokeMethod($ZoneService->,
                      "CreateZoneMembershipSettingData",
                      %InArguments[], %OutArguments[])

if (#status != 0){
    <ERROR! Call to method CreateZoneMembershipSettingData failed
    #status>
}

// 4. Save the returned member objectpath for verification
$ZoneMember-> = %OutArguments["ZoneMembershipSettingData"]

// 5. Add the new zone to the existing zone set
%InArguments["ZoneSet"] = $ZoneSet->
%InArguments["Zone"] = $Zone->
#status = InvokeMethod($ZoneService->, "AddZone", %InArguments[],
                      %OutArguments[])

if (#status != 0){
    <ERROR Call to method AddZone failed>
}

// 6. End Session
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}

// 7. Verify that the zone exists in the zone set
$Zones->[]=associatorNames(
    $ZoneSet->,
    "CIM_MemberOfCollection",
    "CIM_Zone",
    "Collection",
    "Member"
)

// see if the zone is in the returned array
if(!contains($Zone->,$Zones->[])){
    <ERROR! Failed to verify that Zone was added to ZoneSet>
}

```

7.8.5 Create new ZoneSet and add existing Zone

```

// DESCRIPTION
// Create new ZoneSet and add existing Zone
//
// Assume the client has already invoked some logic to determine which
// System (fabric or switch) will host the zone database and zone
// service to be used. Request and obtain a fabric session from the
// zone service. Create a new ZoneSet with a given name, using an

```

```
// extrinsic method. The session may not be ended if any ZoneSet is
// empty, so add an existing zone to the ZoneSet. Then close the
// fabric session.
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. The System hosting the zone database (ComputerSystem or
//     AdminDomain) has been previously identified and defined in the
//     $System-> variable
//
// 2. The name for the new zone set is defined in the #ZoneSetName
//     variable
//
// 3. An existing zone is defined in the $Zone-> variable
//
// FUNCTIONS

// 1. Get the Zone Service and start the session

$ZoneServices->[] = AssociatorNames($System->, "CIM_HostedService",
                                   "CIM_ZoneService", null, null)
// Assumption 1 (above) guarantees there is a zone service for this
// System, Fabric Profile mandates there is no more than one zone
// service for this System
$ZoneService-> = $ZoneServices->[0]

if (!&startSession($ZoneService->)){
    <ERROR! Failed to start zone session>
}

// 2. Create a zone set
%InArguments["ZoneSetName"] = #ZoneSetName
#status = InvokeMethod($ZoneService->, "CreateZoneSet", %InArguments[],
                      %OutArguments[])
if (#status != 0){
    <ERROR! Call to method CreateZoneSet failed>

$ZoneSet-> = %OutArguments["ZoneSet"]

// 3. Add the existing zone to the new zone set
%InArguments["ZoneSet"] = $ZoneSet->
%InArguments["Zone"] = $Zone->
#status = InvokeMethod($ZoneService->, "AddZone", %InArguments[],
                      %OutArguments[])
if (#status != 0){
    <ERROR! Call to method AddZone failed #status>
}

// 4. End Session
if(!&endSession($ZoneService->)){
```

```

        <ERROR! Failed to end zone session, changes may not be committed>
    }

    // 5. Verify that the new zone set exists in the zone database
    try{
        GetInstance($ZoneSet->);
    }catch(CIM_ERR_NOT_FOUND){
        <ERROR! Failed to verify ZoneSet created>
    }

```

7.8.6 Delete zone

```

// DESCRIPTION
// Delete Zone
//
// Try to use intrinsic delete operation to delete a Zone instance.
// Before any operations can be imposed on the zoning service, a
// session is requested and obtained from the zone service. If the
// deletion fails, this may be because the zone is active, or because
// it is not empty. In the latter case, remove all members from the
// zone by deleting the ElementSettingData association instances, and
// try the deletion again.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1. The object name of the zone to be deleted is defined in the
//     $Zone-> variable
// 2. The object name of the zone service object for the System
//     hosting the zone database is defined in the $ZoneService->
//     variable

if(!&startSession($ZoneService->)){
    <ERROR! Failed to start session>
}

try {
    DeleteInstance($Zone->)
}

catch(CIM_ERR_FAILED) {
    // Verify that Zone is not active
    $Zone = GetInstance($Zone->, false, false, false, null)
    if ($Zone.Active) {
        // tell client of its logic problem
        <ERROR! May not delete Zone from active ZoneSet>
    }

    // Failure may be caused because zone has members
    // Try to delete all zone memberships (not zone members themselves)
    $ZoneElements->[] = ReferenceNames($Zone->, "CIM_ElementSettingData",
                                     null)

```

```
for #i in $ZoneElements->[] {
    DeleteInstance($ZoneElements[#i])
}

// Try again
DeleteInstance($Zone->)
}
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not be committed>
}
// Verify that the zone no longer exists in the zone database
try{
    GetInstance($Zone->)
} catch(CIM_ERR_NOT_FOUND){
    // expect failure
    return
}
// error if no exception thrown
<ERROR! Found Zone that should have been deleted>
```

7.8.7 Delete ZoneSet

```
// DESCRIPTION
// Delete Zone Set
//
// Try to use intrinsic delete operation to delete a ZoneSet
// instance. Before any operations can be imposed on the zoning
// service, a session is requested and obtained from the zone service.
// The session is released when the operations are complete. If the
// deletion fails, this may be because the zone set is active, or
// because it is not empty. In the latter case, remove all zones from
// the zone set by deleting the MemberOfCollection association
// instances, and try the deletion again.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1. The object name of the zone set to be deleted is defined in the
//     $ZoneSet-> variable
// 2. The object name of the zone service object for the system
//     hosting the zone database is defined in the $ZoneService->
//     variable

if (!&startSession($ZoneService->))
    <ERROR! Failed to start session>
}

try {
    DeleteInstance($ZoneSet->)
}
catch(CIM_ERR_FAILED) {
```

```

$ZoneSet = GetInstance($ZoneSet->, false, false, false, null)
if ($ZoneSet.Active) {
    // tell client of logic problem
    <ERROR! May not delete an active ZoneSet>
}

// Failure may be because zoneset is not empty
$ZoneMemberships->[] = ReferenceNames($ZoneSet->,
    "CIM_MemberOfCollection", null)
for #i in $ZoneMemberships->[] {
    DeleteInstance($ZoneMemberships->[$i])
}

// Try again
DeleteInstance($ZoneSet->)
}
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}
// Verify that the deletion did indeed occur
try{
    GetInstance($ZoneSet->)
}catch(CIM_ERR_NOT_FOUND){
    // expected, not a recipe error
    return
}
// error if no exception caught
<ERROR! Found ZoneSet that should have been deleted>

```

7.8.8 Delete ZoneMember

```

// DESCRIPTION
// Delete a zone member, removing it from any zones and aliases of
// which it is a member.
//
// Use the intrinsic delete operation to delete a
// ZoneMembershipSettingData instance. Before any operations can be
// imposed on the zoning service, a session is requested and obtained
// from the zone service. The session is released when the operations
// are complete.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTION
// 1. The object name of the ZoneMembershipSettingData to be deleted is
//     defined in the
//     $ZoneMember-> variable
// 2. The object name of the zone service object for the system
//     hosting the zone database is defined in the $ZoneService->
//     variable

```

```

if(!&startSession($ZoneService->)){
    <ERROR! Failed to start session>
}

DeleteInstance($ZoneMember->)
if(!&endSession($ZoneService->)){
    <ERROR! Failed to end session, changes may not have been committed>
}
// verify that it is indeed deleted
try{
    GetInstance($ZoneMember->)
}catch(CIM_ERR_NOT_FOUND){
    // expect an exception,
    // not a recipe error
    return
}
// error if no exception caught
<ERROR! Found ZoneMember that should have been deleted>

```

7.9 Registered Name and Version

Zone Control version 1.4.0 (Component Profile).

7.10 CIM Elements

7.10.1 General

Table 73 describes the CIM elements for Zone Control.

Table 73 - CIM Elements for Zone Control

Element Name	Requirement	Description
7.10.2 CIM_HostedService (Fabric (AdminDomain) to ZoneService)	Optional	Associates the ZoneService to the AdminDomain representing the fabric.
7.10.3 CIM_HostedService (Switch (ComputerSystem) to ZoneService)	Optional	Associates the ZoneService to the ComputerSystem representing the switch.
7.10.4 CIM_ZoneService (Zone Service)	Mandatory	The service that allows for all of the zoning configuration changes.

7.10.2 CIM_HostedService (Fabric (AdminDomain) to ZoneService)

Associates the ZoneService to the AdminDomain representing the fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 74 describes class CIM_HostedService (Fabric (AdminDomain) to ZoneService).

**Table 74 - SMI Referenced Properties/Methods for CIM_HostedService
(Fabric (AdminDomain) to ZoneService)**

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The reference to the AdminDomain representing the fabric (OtherIdentifyingInfo contains 'Fabric' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').
Dependent		Mandatory	The reference to the ZoneService.

7.10.3 CIM_HostedService (Switch (ComputerSystem) to ZoneService)

Associates the ZoneService to the ComputerSystem representing the switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 75 describes class CIM_HostedService (Switch (ComputerSystem) to ZoneService).

**Table 75 - SMI Referenced Properties/Methods for CIM_HostedService
(Switch (ComputerSystem) to ZoneService)**

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The reference to the ComputerSystem representing the switch.
Dependent		Mandatory	The reference to the ZoneService.

7.10.4 CIM_ZoneService (Zone Service)

The service that allows for all of the zoning configuration changes.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 76 describes class CIM_ZoneService (Zone Service).

Table 76 - SMI Referenced Properties/Methods for CIM_ZoneService (Zone Service)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	The Class Name.
Name		Mandatory	Opaque.
OperationalStatus		Mandatory	Status of Zoning Service.
SessionState		Mandatory	State of session. Valid values are "Starting", "Ended".
RequestedSessionState		Mandatory	The requested session state from the client. The valid values that can be set are "Start", "End", and "Terminate".
DefaultZoningState		Mandatory	
CreateZoneSet()		Mandatory	The method creates a ZoneSet and associates it to the System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.
CreateZone()		Mandatory	The method creates a Zone and associates it to System (AdminDomain representing the Fabric or the ComputerSystem representing the Switch) that the ZoneService is hosted on.
CreateZoneMembershipSettingData()		Mandatory	The method creates a ZoneMembershipSettingData (a zone member) and adds it to the specified Zone or NamedAddressCollection representing a Fibre Channel Node.
AddZone()		Mandatory	The method adds to the specified ZoneSet the specified Zone.
AddZoneMembershipSettingData()		Mandatory	The method adds to the specified Zone or NamedAddressCollection representing the Fibre Channel Node the specified ZoneMembershipSettingData (a zone member).
ActivateZoneSet()		Mandatory	The method activates the specified ZoneSet.
SessionControl()		Mandatory	The method enables a client to request a lock of the fabric to begin zoning configuration changes.

STABLE

IMPLEMENTED

8 FDMI Subprofile

8.1 Description

The Fabric-Device Management Interface (FDMI) enables the management of devices such as HBAs through the Fabric. The FDMI complements data in the Fabric Profile. It allows for any entity in the Fabric to expose through SMI the HBA information without having an agent resident on the Host containing the HBA. Figure 10 shows an FDMI instance.

This profile only addresses HBA type devices. The HBA Management Interface defined by FDMI is a subset of interface defined by the Fibre Channel HBA API specification, as explained in ISO/IEC 24775-7 *Information Technology – Storage Management, Part 7 Host Elements*, Clause 6 FC HBA Profile.

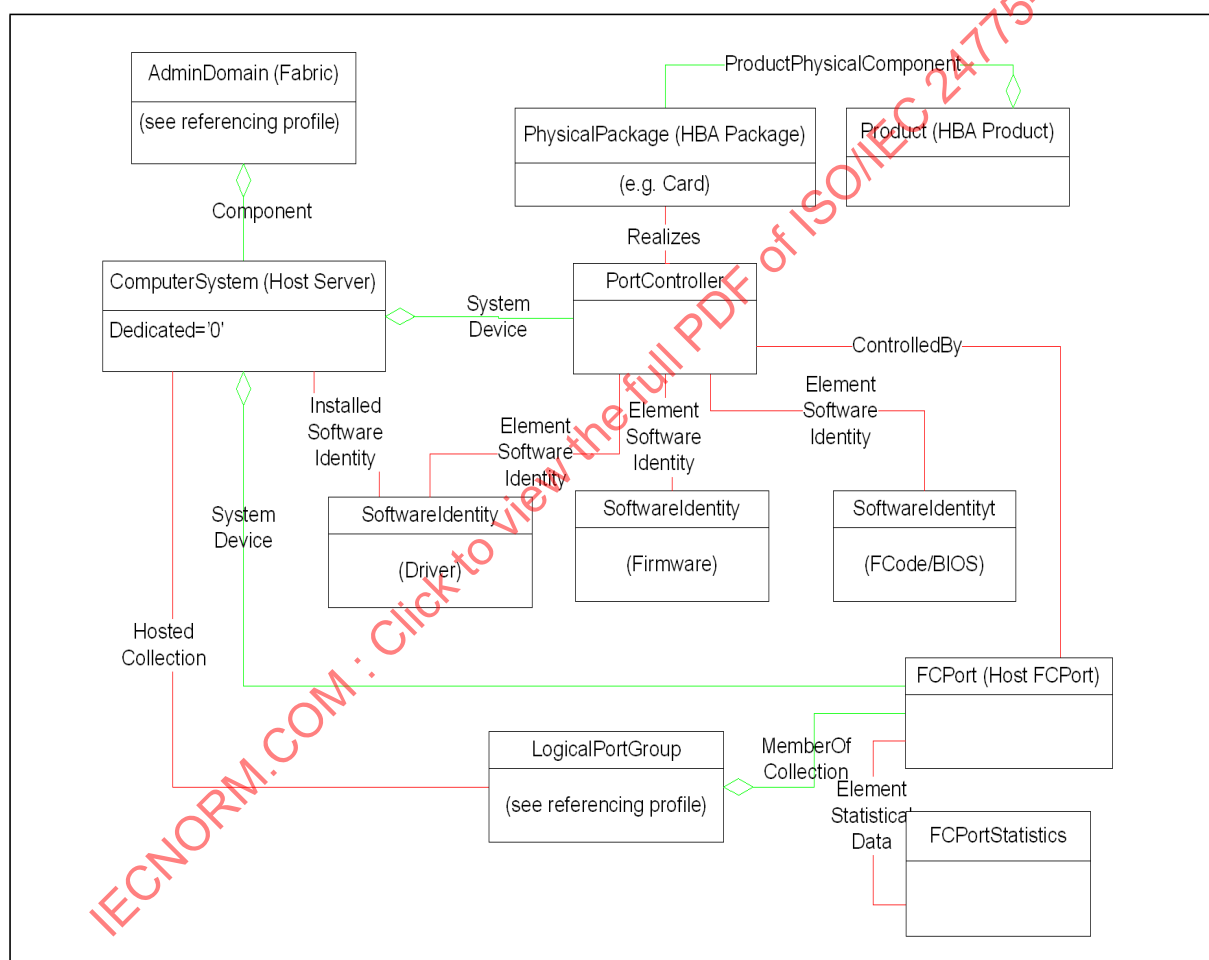


Figure 10 - FDMI Instance

8.2 Health and Fault Management

None.

8.3 Cascading Considerations

None.

8.4 Dependencies on Profiles, Subprofiles, and Packages

Related Profiles for FDMI: Not defined in this International Standard.

8.5 Methods of this Profile

None.

8.6 Client Considerations and Recipes

None.

8.7 Registered Name and Version

FDMI version 1.4.0 (Component Profile).

8.8 CIM Elements

8.8.1 General

Table 77 describes the CIM elements for FDMI.

Table 77 - CIM Elements for FDMI

Element Name	Requirement	Description
8.8.2 CIM_Component (Host Server to Fabric)	Mandatory	Aggregates Hosts (Platforms) in the AdminDomain that represents the Fabric.
8.8.3 CIM_ComputerSystem (Host Server)	Mandatory	The System the HBA is within.
8.8.4 CIM_ControlledBy	Mandatory	Associates the FCPorts of the Host Server with the PortController.
8.8.5 CIM_ElementSoftwareIdentity	Mandatory	Associates the SoftwareIdentity to the HBA.
8.8.6 CIM_FCPort (Host FCPort)	Mandatory	The HBA Fibre Channel Port.
8.8.7 CIM_HostedCollection (System to LogicalPortGroup)	Mandatory	Associates the LogicalPortGroup (Fibre Channel Node) to the hosting System.
8.8.8 CIM_InstalledSoftwareIdentity	Mandatory	Associates the SoftwareIdentity representing the driver to the System it is installed on.
8.8.9 CIM_MemberOfCollection (LogicalPortGroup to FCPort)	Mandatory	Associates FCPort to the LogicalPortGroup.
8.8.10 CIM_PhysicalPackage (HBA Package)	Mandatory	The physical package that the HBA is contained in.
8.8.11 CIM_PortController	Mandatory	The HBA.
8.8.12 CIM_Product (HBA Product)	Mandatory	The product information for the HBA.
8.8.13 CIM_ProductPhysicalComponent	Mandatory	Associates the Product to the PhysicalPackage.
8.8.14 CIM_Realizes	Mandatory	Associates the PhysicalPackage to the PortController.
8.8.15 CIM_SoftwareIdentity (Driver)	Mandatory	The software for the driver.
8.8.16 CIM_SoftwareIdentity (Firmware)	Mandatory	The software for the firmware.
8.8.17 CIM_SoftwareIdentity (Option ROM)	Mandatory	The software for the Option ROM.
8.8.18 CIM_SystemDevice (ComputerSystem to FCPort)	Mandatory	Associates the FCPort to the Host Server ComputerSystem.
8.8.19 CIM_SystemDevice (ComputerSystem to PortController)	Mandatory	Associates the Host Server ComputerSystem with the PortController.

8.8.2 CIM_Component (Host Server to Fabric)

Aggregates Hosts (Platforms) in the AdminDomain that represents the Fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 78 describes class CIM_Component (Host Server to Fabric).

Table 78 - SMI Referenced Properties/Methods for CIM_Component (Host Server to Fabric)

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	Reference to a Host Server ComputerSystem.
GroupComponent		Mandatory	Reference to the AdminDomain representing the Fabric.

8.8.3 CIM_ComputerSystem (Host Server)

The system the HBA is within. It is identified using Host Name from the FDMI interface.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 79 describes class CIM_ComputerSystem (Host Server).

Table 79 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Host Server)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name		Mandatory	The FDMI name of the host containing the Device. The key identifier helping in discovery to determine which HBAs are in the same host.
ElementName		Mandatory	The Platform Label.
NameFormat		Mandatory	
Dedicated		Mandatory	For a FC-GS-6 Platform Type of Host, this shall be 'Not Dedicated' (0).

8.8.4 CIM_ControlledBy

Associates the FCPorts of the Host Server with the PortController.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 80 describes class CIM_ControlledBy.

Table 80 - SMI Referenced Properties/Methods for CIM_ControlledBy

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	A reference to an instance of a Host Server FCPort.
Antecedent		Mandatory	The reference to the PortController.

8.8.5 CIM_ElementSoftwareIdentity

Associates the SoftwareIdentities representing the various software for the HBA to the PortController representing the HBA.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 81 describes class CIM_ElementSoftwareIdentity.

Table 81 - SMI Referenced Properties/Methods for CIM_ElementSoftwareIdentity

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	A reference to an instance of software (Firmware, Driver or Option ROM).
Dependent		Mandatory	A reference to the HBA PortController.

8.8.6 CIM_FCPort (Host FCPort)

The HBA Fibre Channel Port.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 82 describes class CIM_FCPort (Host FCPort).

Table 82 - SMI Referenced Properties/Methods for CIM_FCPort (Host FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.
ElementName		Mandatory	Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.
LinkTechnology		Mandatory	'FC'.
PermanentAddress		Mandatory	Fibre Channel Port WWN.
NetworkAddresses	C	Mandatory	Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 77 for more information about formats).
ActiveFC4Types		Mandatory	The active Fibre Channel FC-4 protocol.
SupportedFC4Types		Optional	An array of integers indicating the Fibre Channel FC-4 protocols supported EXPERIMENTAL When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.

Table 82 - SMI Referenced Properties/Methods for CIM_FCPort (Host FCPort)

Properties	Flags	Requirement	Description and Notes
SupportedCOS		Optional	An array of integers indicating the Fibre Channel Classes of Service that are supported.
PortType		Mandatory	The specific port type currently enabled (from FC-GS-6 Port.Type).
Speed		Optional	Speed of zero represents a link not established. 1 GB is 1 062 500 000 bit/s 2 GB is 2 125 000 000 bit/s 4 GB is 4 250 000 000 bit/s 10 GB single channel variants are 10 518 750 000 bit/s 10 GB four channel variants are 12 750 000 000 bit/s This is the raw bit rate.

8.8.7 CIM_HostedCollection (System to LogicalPortGroup)

Associates the LogicalPortGroup (Fibre Channel Node) to the hosting System.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 83 describes class CIM_HostedCollection (System to LogicalPortGroup).

Table 83 - SMI Referenced Properties/Methods for CIM_HostedCollection (System to LogicalPortGroup)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The Host Server ComputerSystem.
Dependent		Mandatory	The LogicalPortGroup.

8.8.8 CIM_InstalledSoftwareIdentity

Associates the SoftwareIdentity representing the driver to the System it is installed on.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 84 describes class CIM_InstalledSoftwareIdentity.

Table 84 - SMI Referenced Properties/Methods for CIM_InstalledSoftwareIdentity

Properties	Flags	Requirement	Description and Notes
InstalledSoftware		Mandatory	A reference to Driver Software.
System		Mandatory	A reference to the Host Server ComputerSystem.

8.8.9 CIM_MemberOfCollection (LogicalPortGroup to FCPort)

Associates FCPort to the LogicalPortGroup.

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory.

Table 85 describes class CIM_MemberOfCollection (LogicalPortGroup to FCPort).

**Table 85 - SMI Referenced Properties/Methods for CIM_MemberOfCollection
 (LogicalPortGroup to FCPort)**

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	A reference to a LogicalPortGroup on the Host Server.
Member		Mandatory	A reference to an FCPort on the Host Server.

8.8.10 CIM_PhysicalPackage (HBA Package)

The physical package that the HBA is contained by. It can be simply a PhysicalPackage that the system and HBA is contained within. If it is known that the HBA is on a separate board, Card (a subclass of PhysicalPackage) can be used.

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory.

Table 86 describes class CIM_PhysicalPackage (HBA Package).

**Table 86 - SMI Referenced Properties/Methods for CIM_PhysicalPackage
 (HBA Package)**

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Tag		Mandatory	An arbitrary string that uniquely identifies the PhysicalPackage.
ElementName		Optional	User-friendly name. This property is OPTIONAL.
Name		Optional	
Manufacturer		Mandatory	
Model		Mandatory	
SerialNumber		Optional	
Version		Optional	
PartNumber		Optional	

8.8.11 CIM_PortController

The HBA. The HBA may have logical operations that can apply to it (e.g. OperationalStatus).

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory.

Table 87 describes class CIM_PortController.

Table 87 - SMI Referenced Properties/Methods for CIM_PortController

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
ControllerType		Mandatory	

8.8.12 CIM_Product (HBA Product)

The product information for the HBA.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 88 describes class CIM_Product (HBA Product).

Table 88 - SMI Referenced Properties/Methods for CIM_Product (HBA Product)

Properties	Flags	Requirement	Description and Notes
Name		Mandatory	Commonly used Product name.
IdentifyingNumber		Mandatory	Product identification such as a serial number.
Vendor		Mandatory	The manufacturer or the OEM.
Version		Mandatory	Product version information.
ElementName		Mandatory	User-friendly name. Suggested use is Vendor, Version and product name.

8.8.13 CIM_ProductPhysicalComponent

Associates the Product to the PhysicalPackage. This is necessary to link the Product information to the HBA.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 89 describes class CIM_ProductPhysicalComponent.

Table 89 - SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	The product information (CIM_Product) for the HBA Package.
PartComponent		Mandatory	The Physical Package for the HBA.

8.8.14 CIM_Realizes

Associates the PhysicalPackage to the PortController.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 90 describes class CIM_Realizes.

Table 90 - SMI Referenced Properties/Methods for CIM_Realizes

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The Physical Package for the HBA.
Dependent		Mandatory	The PortController of the HBA.

8.8.15 CIM_SoftwareIdentity (Driver)

The software for the driver.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 91 describes class CIM_SoftwareIdentity (Driver).

Table 91 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Driver)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	
VersionString		Mandatory	
Manufacturer		Mandatory	
BuildNumber		Optional	
MajorVersion		Optional	
RevisionNumber		Optional	
MinorVersion		Optional	
Classifications		Mandatory	

8.8.16 CIM_SoftwareIdentity (Firmware)

The software for the firmware.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 92 describes class CIM_SoftwareIdentity (Firmware).

Table 92 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Firmware)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	
VersionString		Mandatory	
Manufacturer		Mandatory	
BuildNumber		Optional	
MajorVersion		Optional	
RevisionNumber		Optional	
MinorVersion		Optional	
Classifications		Mandatory	

8.8.17 CIM_SoftwareIdentity (Option ROM)

The software for the Option ROM.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 93 describes class CIM_SoftwareIdentity (Option ROM).

Table 93 - SMI Referenced Properties/Methods for CIM_SoftwareIdentity (Option ROM)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	
VersionString		Mandatory	
Manufacturer		Mandatory	
BuildNumber		Optional	
MajorVersion		Optional	
RevisionNumber		Optional	
MinorVersion		Optional	
Classifications		Mandatory	

8.8.18 CIM_SystemDevice (ComputerSystem to FCPort)

Associates the FCPort to the Host Server ComputerSystem.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 94 describes class CIM_SystemDevice (ComputerSystem to FCPort).

**Table 94 - SMI Referenced Properties/Methods for CIM_SystemDevice
(ComputerSystem to FCPort)**

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	A reference to a Host FCPort.
GroupComponent		Mandatory	A reference to the Host Server ComputerSystem.

8.8.19 CIM_SystemDevice (ComputerSystem to PortController)

Associates the Host Server ComputerSystem with the PortController.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 95 describes class CIM_SystemDevice (ComputerSystem to PortController).

**Table 95 - SMI Referenced Properties/Methods for CIM_SystemDevice
(ComputerSystem to PortController)**

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	A reference to the HBA PortController.
GroupComponent		Mandatory	A reference to the Host Server ComputerSystem.

IMPLEMENTED

EXPERIMENTAL

9 Fabric Path Performance Subprofile

9.1 Description

The Fabric Path Performance Subprofile extends the standard capabilities of obtaining performance associated to a port to identify the performance in the path defined by an initiator and target ProtocolEndpoint. In the current networking model, the path through the “cloud” is defined by NetworkPipe which is a class that is associated to a ProtocolEndpoint by EndpointOfNetworkPipe. Since the statistics model is defined to allow an association to any LogicalElement, the statistics collected for an NetworkPort, NetworkPortStatistics, can be associated to the NetworkPipe also. When a device supports the Fabric Path Performance Subprofile, it will instantiate the NetworkPipe and as it collects statistics will instantiate the StatisticalData as shown in Figure 11.

The class, StatisticsCollection, provides a mechanism to “collect” all the statistics associated to the NetworkPipes.

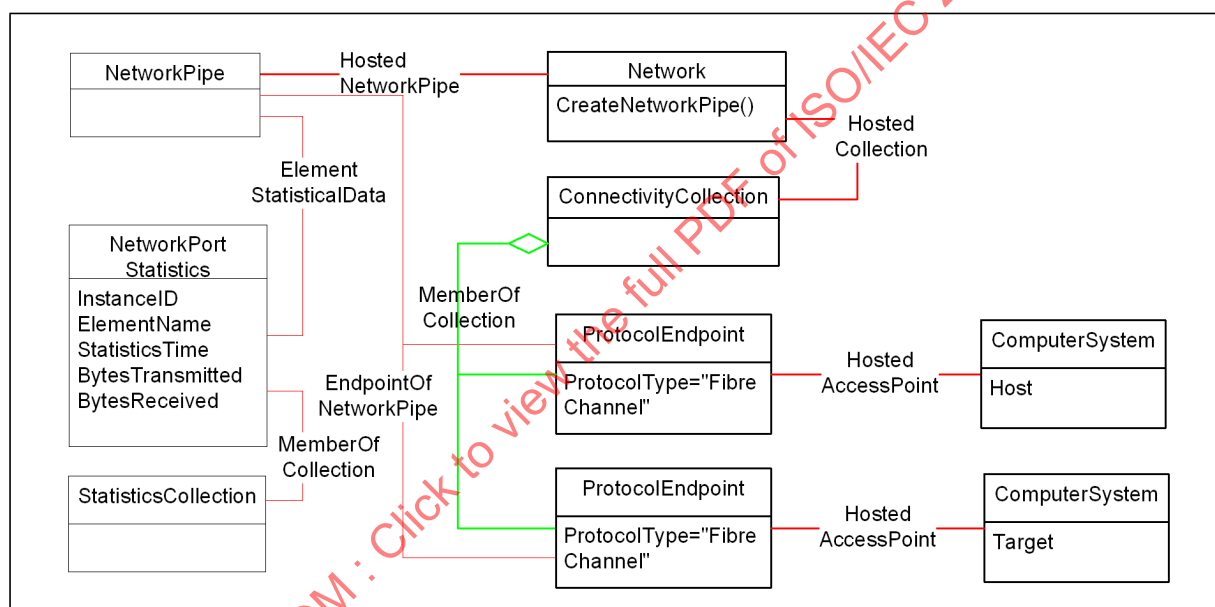


Figure 11 - Instance Diagram

9.2 Health and Fault Management

None.

9.3 Dependencies on Profiles, Subprofiles, and Packages

None.

9.4 Methods of this Profile

The method **CreateNetworkPipes** is used to indicate to the underlying instrumentation that statistics shall be collected between the sets of **ServiceAccessPoints**. A **NetworkPipe** may be created for each set of statistics needed, though the underlying hardware may have maximum limitations. If the creation of the **NetworkPipes** are successful, the client can then discover the **NetworkPipes** and also the associated statics. It is implementation specific how long the statistics are collected. For instance, the **NetworkPipes** may persist as long as the hardware is not rebooted.

```
CreateNetworkPipes (
    [IN] Directionality[],
```

```
[IN] CIM_ServiceAccessPoint REF SourceSAP[],
[IN] CIM_ServiceAccessPoint REF SinkSAP[],
[OUT] CIM_ConcreteJob REF Job,
[OUT] CIM_NetworkPipe REF NetworkPipe[];
```

9.5 Client Considerations and Recipes

None.

9.6 Registered Name and Version

Fabric Path Performance version 1.2.0 (Component Profile).

9.7 CIM Elements

9.7.1 General

Table 96 describes the CIM elements for Fabric Path Performance.

Table 96 - CIM Elements for Fabric Path Performance

Element Name	Requirement	Description
9.7.2 CIM_ElementStatisticalData	Mandatory	Associates FCPortStatistics to the FCPort.
9.7.3 CIM_EndpointOfNetworkPipe	Mandatory	Associates NetworkPipe to the ProtocolEndpoint.
9.7.4 CIM_HostedCollection	Mandatory	Associates the Statistics Collection to the Network representing the fabric.
9.7.5 CIM_HostedNetworkPipe	Mandatory	Associates NetworkPipe to the Network.
9.7.6 CIM_MemberOfCollection	Mandatory	Associates the NetworkPortStatistics to the StatisticsCollection.
9.7.7 CIM_Network	Mandatory	Subclass of AdminDomain representing the fabric.
9.7.8 CIM_NetworkPipe	Mandatory	Pipe through the cloud from an initiator to the target.
9.7.9 CIM_NetworkPortStatistics	Mandatory	NetworkPort Statistics of the NetworkPipe.
9.7.10 CIM_ProtocolEndpoint	Mandatory	The initiator or target (ends of the NetworkPipe).
9.7.11 CIM_StatisticsCollection	Mandatory	Collection to aggregate the NetworkPipe statistics.

9.7.2 CIM_ElementStatisticalData

Associates the NetworkPortStatistics being collected by the monitor to the NetworkPipe (representing the monitor and defining the path being monitored).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 97 describes class CIM_ElementStatisticalData.

Table 97 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	Reference to NetworkPipe representing the monitor that collects the statistics.
Stats		Mandatory	Reference to NetworkPortStatistics that the monitor is collecting.

9.7.3 CIM_EndpointOfNetworkPipe

Associates NetworkPipe representing the monitor to the ProtocolEndpoint. The set of associations (both ends of the NetworkPipe) represent the path that statistics are collected for.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 98 describes class CIM_EndpointOfNetworkPipe.

Table 98 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	Reference to ProtocolEndpoint representing one end of the path to collect statistics for.
Dependent		Mandatory	Reference to NetworkPipe representing the monitor.

9.7.4 CIM_HostedCollection

Associates the Statistics Collection to the Network representing the fabric.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 99 describes class CIM_HostedCollection.

Table 99 - SMI Referenced Properties/Methods for CIM_HostedCollection

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to ConnectivityCollection which collects all ProtocolEndpoints in a fabric.
Antecedent		Mandatory	Reference to Network representing the fabric.

9.7.5 CIM_HostedNetworkPipe

Associates NetworkPipe to the Network representing the fabric. To find all the monitors running in a fabric, this association should be used.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 100 describes class CIM_HostedNetworkPipe.

Table 100 - SMI Referenced Properties/Methods for CIM_HostedNetworkPipe

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	Reference to Network representing fabric.
Dependent		Mandatory	Reference to NetworkPipe representing the monitor that collects the statistics.

9.7.6 CIM_MemberOfCollection

Associates the NetworkPortStatistics to the StatisticsCollection which collects all of the statistics for fabric path performance.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 101 describes class CIM_MemberOfCollection.

Table 101 - SMI Referenced Properties/Methods for CIM_MemberOfCollection

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to StatisticsCollection that collects all statistics that are for fabric path performance.
Member		Mandatory	Reference to the statistics for fabric path performance.

9.7.7 CIM_Network

Subclass of AdminDomain that represents the fabric. NetworkPipes are scoped to the Network via the association HostedNetworkPipe. The Network has the method CreateNetworkPipe() to create the NetworkPipes needed to define monitors within the fabric to collect the path statistics.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 102 describes class CIM_Network.

Table 102 - SMI Referenced Properties/Methods for CIM_Network

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of class.
Name		Mandatory	WWN of Fabric.
NameFormat		Mandatory	"WWN".
NameFormat		Mandatory	
CreateNetworkPipe()		Optional	

9.7.8 CIM_NetworkPipe

The NetworkPipe for this profile is instantiated to provide a mechanism to indicate monitors are in place in the network to collect statistical information. NetworkPortStatistics are associated to the pipe via the association ElementStatisticalData to NetworkPortStatistics and subclasses of NetworkPortStatistics (e.g. FCPortStatistics).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 103 describes class CIM_NetworkPipe.

Table 103 - SMI Referenced Properties/Methods for CIM_NetworkPipe

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	
ElementName		Optional	

9.7.9 CIM_NetworkPortStatistics

Network Port Statistics represent a snapshot of counters for the NetworkPipe. An instance of this class can represent the statistics for the current statistics, archived and consolidated statistics, or both.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 104 describes class CIM_NetworkPortStatistics.

Table 104 - SMI Referenced Properties/Methods for CIM_NetworkPortStatistics

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Optional	
StatisticTime		Optional	The time the statistics were collected.
BytesTransmitted		Mandatory	The total number of bytes that are transmitted, including framing characters.
BytesReceived		Mandatory	The total number of bytes that are received, including framing characters.

9.7.10 CIM_ProtocolEndpoint

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 105 describes class CIM_ProtocolEndpoint.

Table 105 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
Name		Mandatory	
NameFormat		Mandatory	
ProtocolIFType		Mandatory	

9.7.11 CIM_StatisticsCollection

The collection, hosted to the Network representing the Fabric, is used to aggregate all the statistics for the NetworkPipes.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 106 describes class CIM_StatisticsCollection.

Table 106 - SMI Referenced Properties/Methods for CIM_StatisticsCollection

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	
SampleInterval		Mandatory	The interval the statistics associated to this collection are collected. This value indicates to clients the minimum period statistics should be sampled.
TimeLastSampled		Mandatory	

EXPERIMENTAL

EXPERIMENTAL

10 Fibre Channel Security Subprofile

10.1 Description

Fibre Channel Security Subprofile describes the Fibre Channel Security Protocol (FC-SP) management policies. This profile is a specialization of the Authorization Subprofile.

The FC-SP Policies can be broken down into four parts:

- Membership
- Connectivity
- Management Access
- Attributes

FC-SP Membership policy is composed of two components, Switch Membership and Device Membership. Switch Membership is fabric-wide information which defines which switches are allowed to be part of a fabric, controls physical management access, and allows switch characteristics to be specified. Device Membership is fabric-wide information which defines which devices are allowed to be part of a fabric, controls in-band management access, and allows device characteristics to be specified. FC-SP Connectivity policy defines connectivity restrictions on a per-switch basis.

This subprofile describes exposing the FC-SP Membership and Connectivity policies. It does not define physical management and switch characteristics as part of the Membership policy. The specialization of the Authorization Subprofile primarily includes the subclassing of Identity to StorageHardwareID to identify the AuthorizedSubject and AuthorizedTargets.

For the Membership Policy, the AuthorizedTarget is always the Fabric identified by its Fabric Name and the two possible AuthorizedSubjects are the switch identified by its switch name (WWN) and a device by its node WWN. In Figure 12, the diagram shows both a switch having membership and a device having membership.

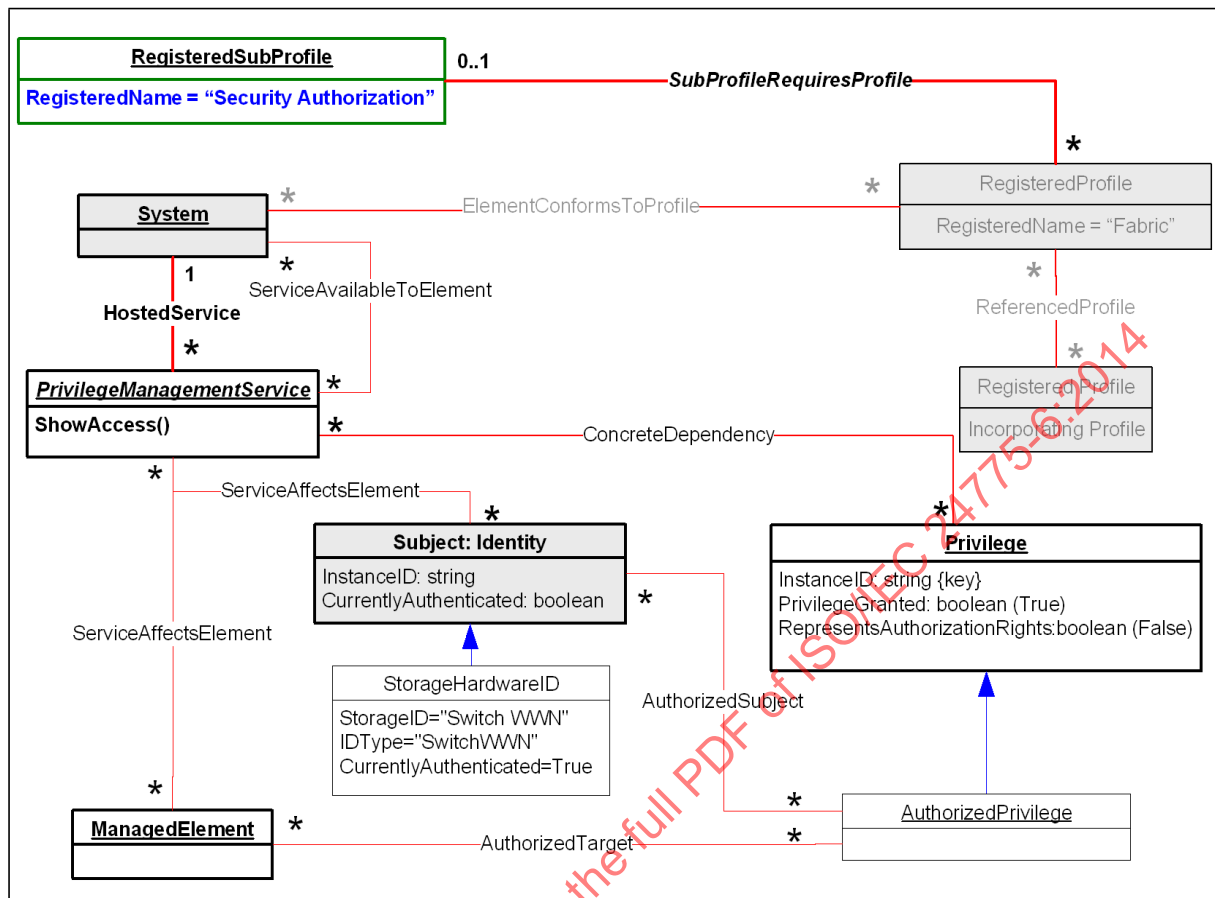


Figure 12 - Specialization of Security Authorization Subprofile for Membership Policy

Figure 13 shows both a switch having Switch Membership and a device having Device Membership. A switch is identified by its switch name (WWN) and a device by its node WWN.

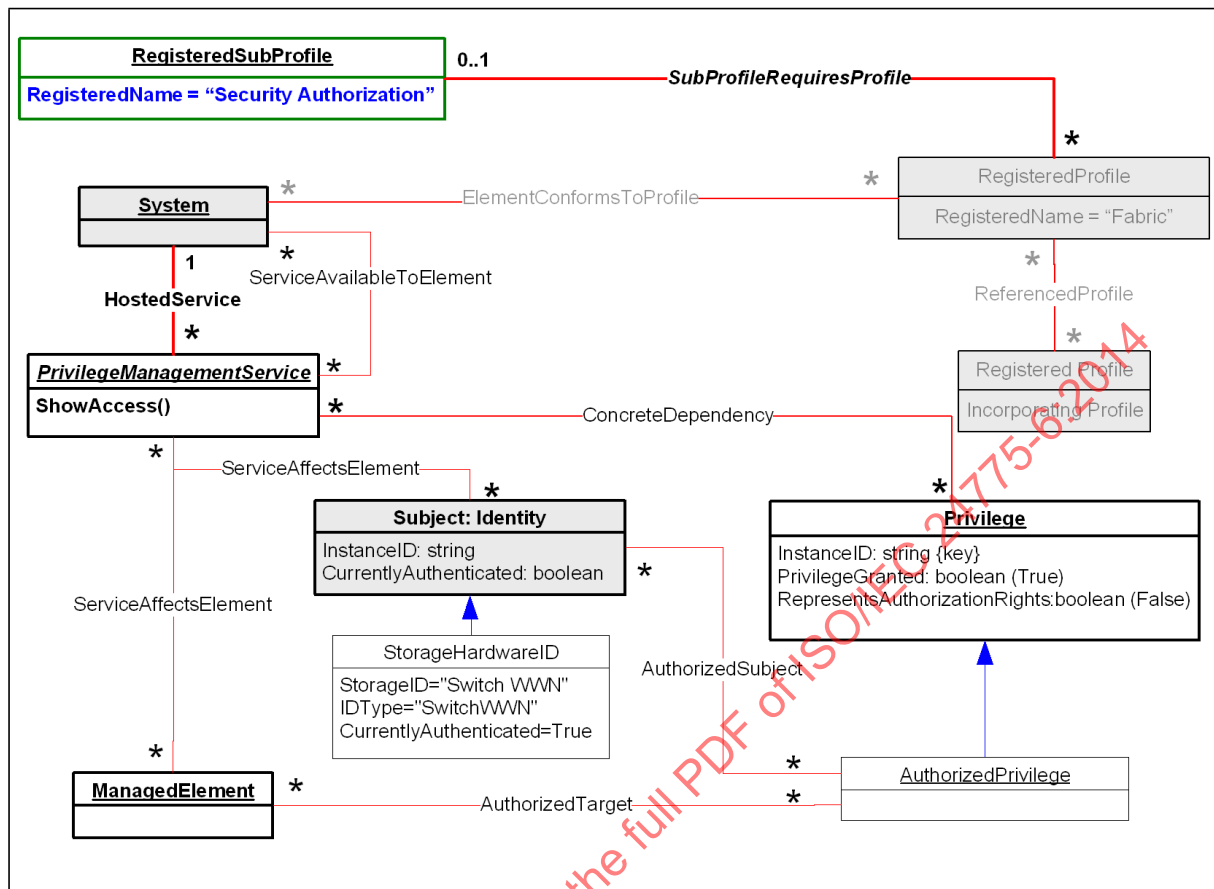


Figure 13 - Specialization of Security Authorization Subprofile for Connectivity Policy

10.2 Health and Fault Management Consideration

None.

10.3 Cascading Considerations

None.

10.4 Supported Profiles, Subprofiles, and Packages

None.

10.5 Methods of the Profile

None.

10.6 Client Considerations and Recipes

None.

10.7 Registered Name and Version

FabricSecurity version 1.2.0 (Component Profile).

10.8 CIM Elements

10.8.1 General

Table 107 describes the CIM elements for FabricSecurity.

Table 107 - CIM Elements for FabricSecurity

Element Name	Requirement	Description
10.8.2 CIM_AuthorizationService	Mandatory	The service controlling the security policy in the fabric.
10.8.3 CIM_AuthorizedPrivilege	Mandatory	Privilege granted to the Switch or the Node.
10.8.4 CIM_AuthorizedSubject	Mandatory	The association of the Node or Switch to its Privilege.
10.8.5 CIM_AuthorizedTarget	Mandatory	The association of the privilege (or a switch or node) to the target (switch or the fabric) that the switch or node is being granted access to.
10.8.6 CIM_HostedService	Mandatory	Associates the AuthorizationService to the AdminDomain that is hosting it.
10.8.7 CIM_ServiceAffectsElement (ManagedElement to Service)	Mandatory	Service Affects Managed Element.
10.8.8 CIM_ServiceAffectsElement (StorageHardwareID to Service)	Mandatory	Service affects StorageHardwareID.
10.8.9 CIM_ServiceAvailableToElement (Fabric AdminDomain to Service)	Mandatory	Service available to fabric.
10.8.10 CIM_StorageHardwareID	Mandatory	The class that identifies the subject to be granted access.

10.8.2 CIM_AuthorizationService

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory

Table 108 describes class CIM_AuthorizationService.

Table 108 - SMI Referenced Properties/Methods for CIM_AuthorizationService

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	Name of System class.
SystemName		Mandatory	Name of class.
CreationClassName		Mandatory	Name of AuthorizationService class.
Name		Mandatory	Uniquely identifies the service in the fabric.

10.8.3 CIM_AuthorizedPrivilege

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 109 describes class CIM_AuthorizedPrivilege.

Table 109 - SMI Referenced Properties/Methods for CIM_AuthorizedPrivilege

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
PrivilegeGranted		Mandatory	
RepresentsAuthorizationRights		Mandatory	

10.8.4 CIM_AuthorizedSubject

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 110 describes class CIM_AuthorizedSubject.

Table 110 - SMI Referenced Properties/Methods for CIM_AuthorizedSubject

Properties	Flags	Requirement	Description and Notes
PrivilegedElement		Mandatory	
Privilege		Mandatory	

10.8.5 CIM_AuthorizedTarget

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 111 describes class CIM_AuthorizedTarget.

Table 111 - SMI Referenced Properties/Methods for CIM_AuthorizedTarget

Properties	Flags	Requirement	Description and Notes
Privilege		Mandatory	
TargetElement		Mandatory	Reference to target element.

10.8.6 CIM_HostedService

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 112 describes class CIM_HostedService.

Table 112 - SMI Referenced Properties/Methods for CIM_HostedService

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	AdminDomain representing the Fabric.
Dependent		Mandatory	AuthorizationService.

10.8.7 CIM_ServiceAffectsElement (ManagedElement to Service)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 113 describes class CIM_ServiceAffectsElement (ManagedElement to Service).

Table 113 - SMI Referenced Properties/Methods for CIM_ServiceAffectsElement (ManagedElement to Service)

Properties	Flags	Requirement	Description and Notes
AffectingElement		Mandatory	AuthorizationService.
AffectedElement		Mandatory	AdminDomain representing the Switch.

10.8.8 CIM_ServiceAffectsElement (StorageHardwareID to Service)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 114 describes class CIM_ServiceAffectsElement (StorageHardwareID to Service).

Table 114 - SMI Referenced Properties/Methods for CIM_ServiceAffectsElement (StorageHardwareID to Service)

Properties	Flags	Requirement	Description and Notes
AffectingElement		Mandatory	AuthorizationService.
AffectedElement		Mandatory	AdminDomain representing the Switch.

10.8.9 CIM_ServiceAvailableToElement (Fabric AdminDomain to Service)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 115 describes class CIM_ServiceAvailableToElement (Fabric AdminDomain to Service).

**Table 115 - SMI Referenced Properties/Methods for CIM_ServiceAvailableToElement
(Fabric AdminDomain to Service)**

Properties	Flags	Requirement	Description and Notes
ServiceProvided		Mandatory	AuthorizationService.
UserOfService		Mandatory	AdminDomain representing the Fabric.

10.8.10 CIM_StorageHardwareID

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 116 describes class CIM_StorageHardwareID.

Table 116 - SMI Referenced Properties/Methods for CIM_StorageHardwareID

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
StorageID		Mandatory	Node WWN, Switch WWN, Port WWN.
IDType		Mandatory	'NodeWWN' 'SwitchWWN' or PortWWN'.
CurrentlyAuthenticated		Mandatory	

EXPERIMENTAL

EXPERIMENTAL

11 Fabric Views Subprofile

11.1 Description

11.1.1 Synopsis

11.1.2 Overview

This Profile specifies SNIA_ View Classes for the Fabric Profiles.

In this release of SMI-S, SNIA_ view classes provide an optimization of retrieval of information provided by multiple (associated) instances in a Profile. There is no support for update of SNIA_ view classes instances. Update of a SNIA_ view class instance can only be accomplished by updating the base class instances from which the view is derived.

11.1.2.1 Goals of SNIA_ View Classes

11.1.2.1.1 Intended Goals

Goals that SNIA_ View Classes are intended to address are:

- Get more data in one call to CIM Server.

The CIM model for fabrics involve a number of classes and associations. The objective is to allow discovery of the fabric model using SNIA_ View Classes with a reduction in the number of association traversals required.

- Allow providers to optimize the Request.

In many cases, the data represented by a View Class is actually kept (and returned) by a device as one entity. When the "normalized" CIM model is traversed many calls are made to retrieve that one entity. The provider takes the data from the one entity and carves it up for each CIM request. In many cases this involves retrieving the same entity multiple times. The objective is to allow a Provider to return the single entity in one SMI-S request (for data that is typically kept together by the device).

11.1.2.1.2 Additional Goals

- Do more things in one call to CIM Server.

An example would be retrieval or discovery of model information with fewer calls. However, this goal also extends to updating the CIM model (e.g., configuration actions). The SNIA_ View Classes are not intended to help in the latter case. However, SNIA_ View Classes should facilitate access to underlying classes in support of configuration operations.

It is important to note that the SNIA_ View Classes proposal was based directly on experiences relating to the scalability and performance of SMI-S real-world implementations. The focus is on improving performance in large configurations (e.g., thousands of volumes and thousands of disk drives).

11.1.2.2 Specific Requirements and Objectives of View Classes

11.1.2.2.1 Pre-defined View Classes

In order to gain the desired performance advantage, it is felt that view classes would have to be pre-defined (in SMI-S) to allow provider optimization of the requested information.

- Enable Associator Calls to View Class instances.

It should be possible to retrieve a View Class by an associator's call to the class.

However, it is desired that the association should be clearly distinguished from existing associations on the base classes.

- Enable Associator Calls from View Class instances.

It should be possible to get related classes (e.g., base classes) from the View Class by using associator calls.

Again, the associations used should be clearly distinguished from existing associations on the base classes.

- Allow View Classes to be used where other classes would be used.

This certainly includes "read" intrinsics as parameters of Extrinsics.

However, at this time "Write" intrinsic support is deferred and use in Extrinsics (as IN or OUT parameters) is not covered in this release of SMI-S.

11.1.2.2.2 Support Life Cycle Indications on View Classes

This requirement is being deferred for consideration in a future release of SMI-S.

11.1.2.3 View Class Capabilities

The implementation shall identify which view classes are implemented using a set of conditions. The model for determining whether or not the Fabric Views Profile is supported and which views are supported is illustrated in Figure 14.

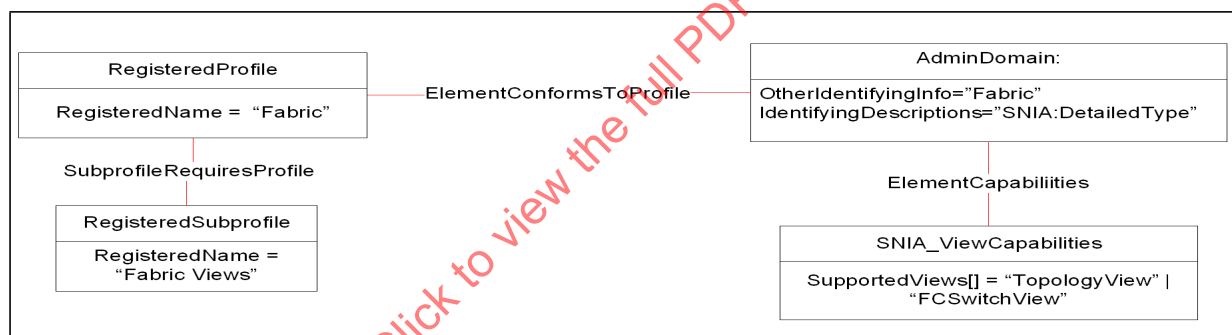


Figure 14 - Fabric View Class Capabilities

First a client may determine whether or not a profile implementation has implemented any view classes by looking for a `RegisteredSubprofile` with a `RegisteredName` of "Fabric Views". If this `RegisteredSubprofile` exists then the profile supports some number of view classes.

Next a client would be able to determine which view classes are supported by an implementation by following the `ElementConformsToProfile` to the Fabric and then following the `ElementCapabilities` from the Fabric to the `SNIA_ViewCapabilities` instance. There shall be one instance of the `SNIA_ViewCapabilities` class for each Fabric if the profile supports the Fabric Views Subprofile. The `SNIA_ViewCapabilities` instance shall have an array of strings (`SupportedViews`) that identify the view classes that are supported. For example, if the `SupportedViews` array includes the "TopologyView" string, then the `TopologyView` class shall be supported.

11.1.3 Topology View

Discovering a topology for a large fibre channel fabric (thousands of ports) time-wise is expensive. The existing fabric profile requires traversals of three associations and two classes for each link. Using the view class reduces this to one enumeration of the view class itself for all links. The `SNIA_TopologyView`, shown in Figure 15, provides the properties to be able to

build a topology without having to do associators from the ProtocolEndpoint to the FCPort to the FCNode or FCSwitch.

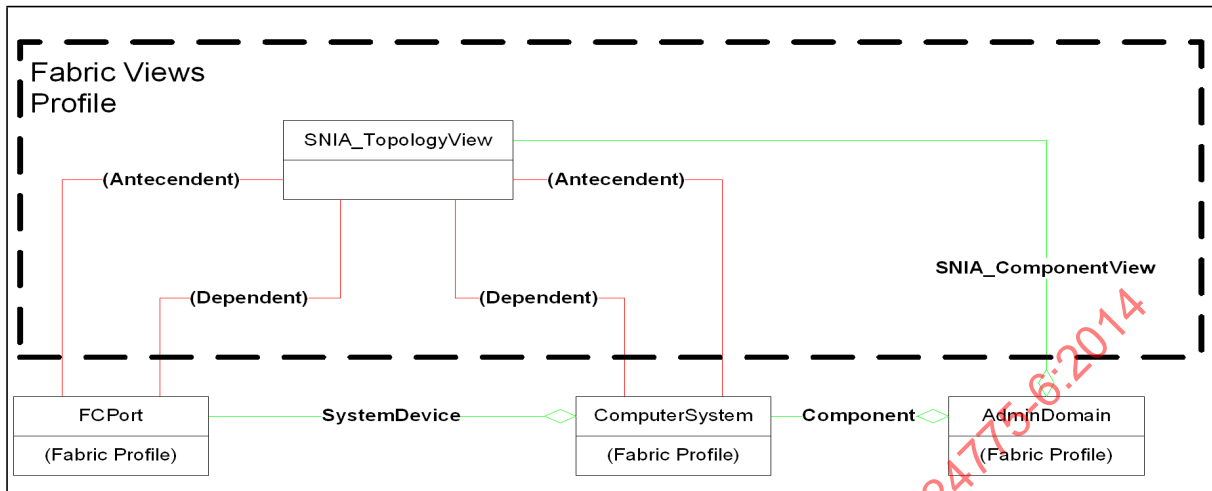


Figure 15 - Topology View Class

11.1.4 FCSwitch View

Discovering a fibre channel switch and its associated ports is expensive. The **SNIA_SwitchView**, shown in Figure 15, provides the properties to be able to discover all the switch properties for a Fibre Channel Switch and Ports without having to do associators from the **ComputerSystem** representing the Switch to **FCPort**, **LogicalModule**, **Product**, **PhysicalPackage**, **SoftwareIdentity**, **AccessPoint**, and **Location**.

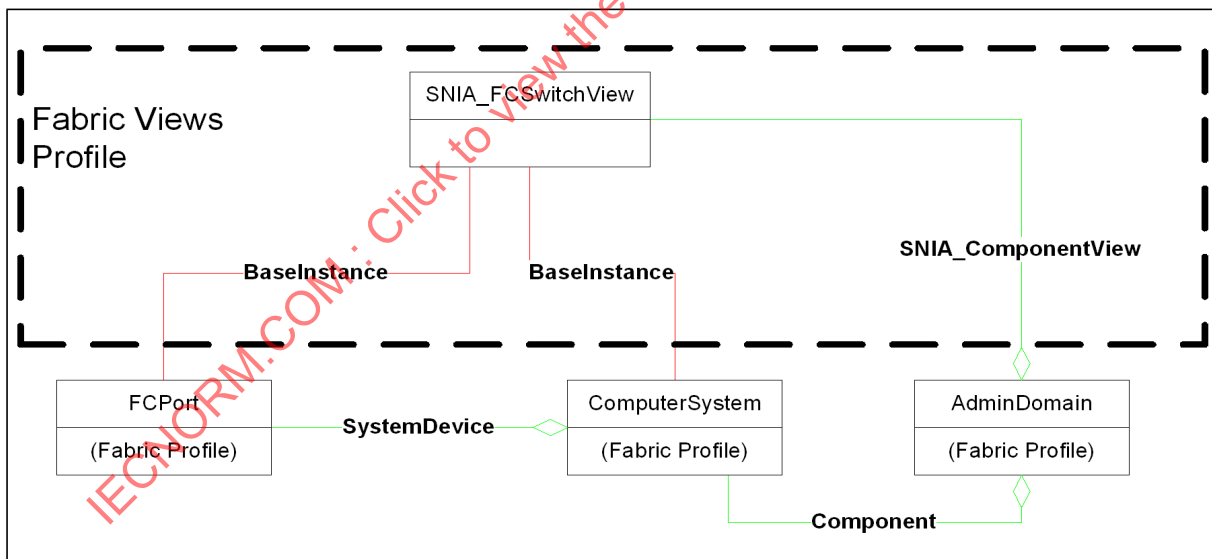


Figure 16 - FCSwitch View Class

11.2 Health and Fault Management Consideration

None.

11.3 Cascading Considerations

None.

11.4 Supported Profiles, Subprofiles, and Packages

Related Profiles for Fabric Views: Not defined in this International Standard.

11.5 Methods of the Profile

None.

11.6 Client Considerations and Recipes

None.

11.7 Registered Name and Version

Fabric Views version 1.5.0 (Component Profile)

11.8 CIM Elements**11.8.1 General**

Table 117 describes the CIM elements for Fabric Views.

Table 117 - CIM Elements for Fabric Views

Element Name	Requirement	Description
11.8.2 CIM_ElementCapabilities (View Capabilities)	Mandatory	Associates the Fabric AdminDomain to the SNIA_ViewCapabilities supported by the implementation.
11.8.3 SNIA_BaseInstance (View to FC Port)	Conditional	Conditional requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'. Associates the SNIA_SwitchView instances to the corresponding FC Port instance.
11.8.4 SNIA_BaseInstance (View to Switch)	Conditional	Conditional requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'. Associates the SNIA_SwitchView instances to the corresponding Fibre Channel Switch instance.
11.8.5 SNIA_ComponentView (FCSwitchView to Fabric)	Conditional	Conditional requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'. Associates the Fabric AdminDomain to the SNIA_FCSwitchView.
11.8.6 SNIA_ComponentView (TopologyView to Fabric)	Conditional	Conditional requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'TopologyView'. Associates the Fabric AdminDomain to the SNIA_TopologyView.

Table 117 - CIM Elements for Fabric Views

Element Name	Requirement	Description
11.8.7 SNIA_FCSwitchView	Conditional	Conditional requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'. The SNIA_SwitchView provides the properties for a Fibre Channel Switch.
11.8.8 SNIA_TopologyView	Conditional	Conditional requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'TopologyView'. The SNIA_TopologyView provides the properties to be able to build a topology for a Fibre Channel Fabric.
11.8.9 SNIA_ViewCapabilities	Mandatory	The SNIA_ViewCapabilities identifies the capabilities of the implementation of view classes.

11.8.2 CIM_ElementCapabilities (View Capabilities)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 118 describes class CIM_ElementCapabilities (View Capabilities).

Table 118 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (View Capabilities)

Properties	Flags	Requirement	Description and Notes
Capabilities		Mandatory	The ViewCapabilities.
ManagedElement		Mandatory	The Fabric AdminDomain that has the ViewCapabilities.

11.8.3 SNIA_BaseInstance (View to FC Port)

Associates the SNIA_SwitchView instances to the corresponding FC Port instance.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'.

Table 119 describes class SNIA_BaseInstance (View to FC Port).

Table 119 - SMI Referenced Properties/Methods for SNIA_BaseInstance (View to FC Port)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The base CIM_FCPort instance on which the SNIA_SwitchView instance is based.
Dependent		Mandatory	The SNIA_FCSwitchView instance that is based on the CIM_FCPort instance.

11.8.4 SNIA_BaseInstance (View to Switch)

Associates the SNIA_SwitchView instances to correspond to the Fibre Channel Switch instance.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'.

Table 120 describes class SNIA_BaseInstance (View to Switch).

Table 120 - SMI Referenced Properties/Methods for SNIA_BaseInstance (View to Switch)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The base CIM_ComputerSystem (Switch) instance on which the SNIA_SwitchView instance is based.
Dependent		Mandatory	The SNIA_FCSwitchView instance that is based on the CIM_ComputerSystem (Switch) instance.

11.8.5 SNIA_ComponentView (FCSwitchView to Fabric)

Associates the Fabric AdminDomain to the SNIA_FCSwitchView.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'.

Table 121 describes class SNIA_ComponentView (FCSwitchView to Fabric).

Table 121 - SMI Referenced Properties/Methods for SNIA_ComponentView (FCSwitchView to Fabric)

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	The Fabric the SwitchView instance belongs to.
PartComponent		Mandatory	The FCSwitchView instance.

11.8.6 SNIA_ComponentView (TopologyView to Fabric)

Associates the Fabric AdminDomain to the SNIA_TopologyView. Note that this is an association (SNIA_ComponentView) to an association (SNIA_TopologyView).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'TopologyView'.

Table 122 describes class SNIA_ComponentView (TopologyView to Fabric).

Table 122 - SMI Referenced Properties/Methods for SNIA_ComponentView (TopologyView to Fabric)

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	The Fabric the TopologyView instance belongs to.
PartComponent		Mandatory	The TopologyView instance.

11.8.7 SNIA_FCSwitchView

The SNIA_SwitchView provides the properties to be able to discover all the switch properties for a Fibre Channel Switch and Ports without having to do associators from the ComputerSystem representing the Switch to FCPort, LogicalModule, Product, PhysicalPackage, SoftwareIdentity, AccessPoint, and Location.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'FCSwitchView'.

Table 123 describes class SNIA_FCSwitchView.

Table 123 - SMI Referenced Properties/Methods for SNIA_FCSwitchView

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
SwitchWWN	C	Mandatory	A Name_Identifier that is associated with a Fibre Channel switch.
DomainID		Mandatory	The Domain ID of the Switch.
ElementName		Optional	The Symbolic Name of the Switch.
OperationalStatus		Mandatory	The operational status of the Switch.
SwitchEnabledState		Mandatory	EnabledState indicates the enabled and disabled states of the Fibre Channel Switch.
SwitchNPIVSupported		Mandatory	Flag indicating whether NPIV is support on the fibre channel switch.
FCPortWWN	C	Mandatory	The FC Port Name Identifier (FC Port WWN).
FCPortElementName		Mandatory	The user-friendly name of the FCPort.
FCPortType		Mandatory	The PortType of the FCPort.).
FCPortOperationalStatus		Mandatory	The operational status of the FCPort.
FCPortEnabledState		Mandatory	EnabledState indicates the enabled and disabled states of the Fibre Channel Port.
PortNumber		Mandatory	The port number labeled for administrative purposes relative to the logical module. This is typically what is exposed through the devices element manager or CLI.
LogicalModulePortNumber		Mandatory	The port number labeled for administrative purposes relative to the logical module.
DomainPortNumber		Mandatory	The port number used for domain id and physical port zone members.

Table 123 - SMI Referenced Properties/Methods for SNIA_FCSwitchView

Properties	Flags	Requirement	Description and Notes
ModuleNumber		Mandatory	Logical modules are often named by the physical or logical slot that they occupy within the Switch.
PortNPIVSupported		Mandatory	Flag indicating whether NPIV is supported on the fibre channel port.
MaxPortNPIVLogins		Mandatory	Maximum number of NPIV logins supported on the fibre channel port.
Vendor		Mandatory	The name of the Product's supplier of the Switch.
ProductName		Mandatory	Commonly used Product name.
VersionString		Mandatory	A string representing the complete software version of the Switch.
Model		Mandatory	
SerialNumber		Mandatory	
PartNumber		Mandatory	
OEM		Mandatory	
OEMProductName		Mandatory	
OEMIdentifyingNumber		Mandatory	
AccessInfo		Mandatory	
LocationName		Mandatory	
LocationPhysicalPosition		Mandatory	

11.8.8 SNIA_TopologyView

The SNIA_TopologyView provides the properties to be able to build a topology for a Fibre Channel fabric without having to do associators from the ProtocolEndpoint to the FCPort to the FCNode or FCSwitch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the array property SNIA_ViewCapabilities.SupportedViews contains the string 'TopologyView'.

Table 124 describes class SNIA_TopologyView.

Table 124 - SMI Referenced Properties/Methods for SNIA_TopologyView

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
AntecedentFCPortWWN	C	Mandatory	The FC Port Name Identifier (FC Port WWN) of the FCPort the antecedent reference is associated to.
AntecedentElementWWN	C	Mandatory	The FC Switch Name Identifier (FC Switch WWN) or FC Node Name (FC Node WNN) the AntecedentFCPort is associated to.
AntecedentFCPortElementName		Mandatory	The user-friendly name of the FCPort the antecedent reference is associated to.
AntecedentFCPortType		Mandatory	The PortType of the FCPort the antecedent reference is associated to.).

Table 124 - SMI Referenced Properties/Methods for SNIA_TopologyView

Properties	Flags	Requirement	Description and Notes
DependentFCPortWWN	C	Mandatory	The FC Port Name Identifier (FC Port WWN) of the FCPort the Dependent reference is associated to.
DependentElementWWN	C	Mandatory	The FC Switch Name Identifier (FC Switch WWN) or FC Node Name (FC Node WNN) the DependentFCPort is associated to.
DependentFCPortElementName		Mandatory	The user-friendly name of the FCPort the dependent reference is associated to.
DependentFCPortType		Mandatory	The PortType of the FCPort the dependent reference is associated to.).
AntecedentFCPort		Mandatory	The reference to the antecedent FCPort that is communicating with the dependent FCPort.
AntecedentSystem		Mandatory	The reference to the system of the antecedent FCPort in the AntecedentFCPort SystemDevice association.
DependentFCPort		Mandatory	The reference to the dependent FCPort that is communicating with the antecedent FCPort.
DependentSystem		Mandatory	The reference to the system of the Dependent FCPort in the DependentFCPort SystemDevice association.

11.8.9 SNIA_ViewCapabilities

The SNIA_ViewCapabilities instance defines the capabilities of an implementation support for SNIA_view classes. The SNIA_ViewCapabilities is subclassed from CIM_Capabilities.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 125 describes class SNIA_ViewCapabilities.

Table 125 - SMI Referenced Properties/Methods for SNIA_ViewCapabilities

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	An opaque, unique id for the view class capability of an implementation.
ElementName		Optional	A provider supplied user-friendly Name for this SNIA_ViewCapabilities element.
SupportedViews		Mandatory	This array of strings lists the view classes that are supported by the implementation. Supported string values are 'TopologyView' or 'FCSwitchView'.

EXPERIMENTAL

EXPERIMENTAL

12 Virtual Fabrics Subprofile

12.1 Description

Technologies have recently been implemented for Fibre Channel SANs to logically separate the hardware into multiple fabrics yet keep them physically interconnected. The term for this technology defined by FC-SW-5 is "Virtual Fabrics". FC-SW-5 calls the underlying hardware "Core Switches". To be consistent with a more broader use model and DMTF schematics, this subprofile will call these the partitioning systems. FC-SW-5 calls the switching construct that resides in the partitioning system, a virtual switch. Again for the broader use model and consistency with the DMTF, this system is called the partitioned system. The Fabric Profile already provides the discovery of the "virtual fabrics" and the "partitioned switches". This subprofile allows for the discovery of the underlying partitioning system. The Switch Partitioning Subprofile provides the mechanism for configuring the partitioning system.

All Virtual Fabrics shall be associated to the same SAN for the same underlying "physical" topology. Also the Registered Subprofile associates to the SAN object to allow the client to arrive at the proper SAN object without traversing upwards from the fabric instance. In Figure 17, the relationship is shown. Note also that the partitioning ComputerSystems are associated with the SAN.

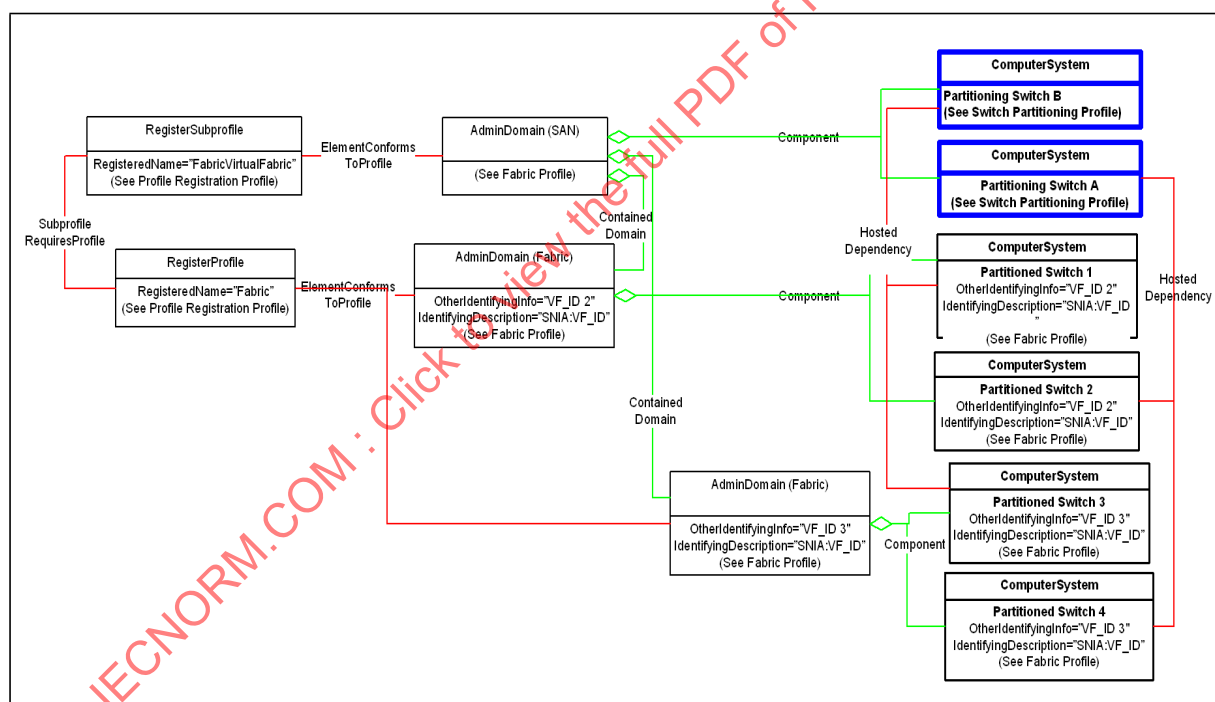


Figure 17 - RegisteredProfile/Subprofile, AdminDomain, and ComputerSystem Relationships

In Figure 17 the only thing added to the model beyond those elements in the Fabric Profile and the Switch Partitioning Profile are the *ElementConformsToProfile* association from the SAN AdminDomain (in the Fabric Profile) to the Virtual Fabrics RegisteredProfile and the *Component* association from the Partitioning ComputerSystem (from the Switch Partitioning Profile) to the same SAN AdminDomain. The Virtual Fabrics Profile is a component profile of the Fabric Profile and it requires implementation of the Switch Partitioning Profile.

In Figure 18, the partitioning systems are shown presenting two "virtual fabrics" with only one link (ActiveConnection) in each fabric. See 5.1.3 for more information regarding fabric topology using ActiveConnection. In this configuration each fabric, partitioned switch, and port are uniquely identified. Note that the new instances to support this subprofile are outlined in a bolded line.

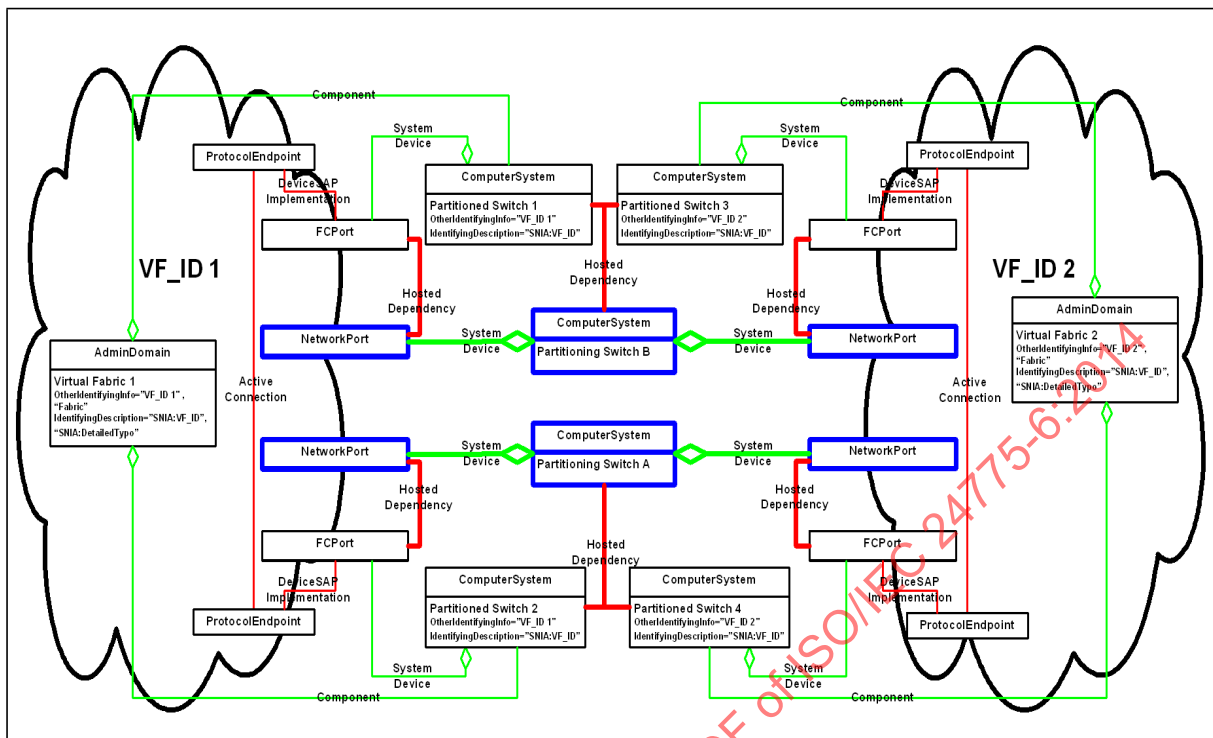


Figure 18 - Two Virtual Fabric and Two Partitioning Systems

In Figure 18, a single partitioning system is creating two virtual fabrics and in Figure 19, the system is partitioning the network port. When the NetworkPort is partitioned, each port is not guaranteed to be uniquely identified (though the key to the instance is) and the client needs to map it to the fabric it is participating in to maintain the ports identity.

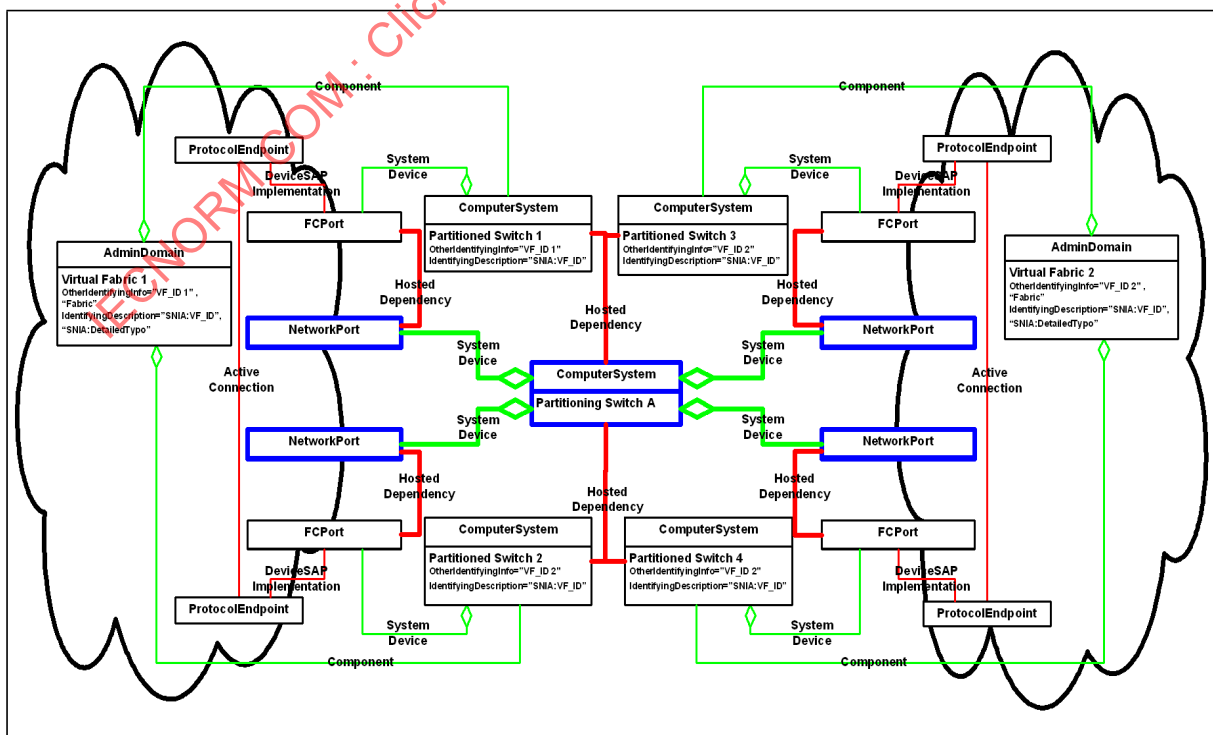


Figure 19 - Two Virtual Fabrics and One Partitioning System

In all cases it is possible to have a one-to-one or many-to-one relationship between partitioned switches and the partitioning switch. The same is true between the partitioned FCPorts and the NetworkPorts.

To determine the end-to-end path of a fabric, the client follows the rules outlined in the Fabric Profile (see 5.1.3) with the exception that when entering a switch, if there exists a hosted dependency on the switch port, the partitioning system shall be traversed instead of the partitioned system. So in Figure 19, when the client traverses across ActiveConnection to the ProtocolEndpoint to the FCPort and down the HostedDependency, it realizes that both ends of the link are actually in the same partitioned system.

12.2 Health and Fault Management Consideration

The possible Health and Fault information through LifeCycle indications are addressed in the Fabric Profile.

12.3 Cascading Considerations

None.

12.4 Supported Profiles, Subprofiles, and Packages

Table 126 describes the supported profiles for FabricVirtualFabrics.

Table 126 - Supported Profiles for FabricVirtualFabrics

Profile Name	Organization	Version	Requirement	Description
FabricSwitchPartitioning	SNIA	1.5.0	Mandatory	

12.5 Methods of the Profile

None.

12.6 Client Considerations and Recipes

None.

12.7 Registered Name and Version

FabricVirtualFabrics version 1.5.0 (Component Profile).

12.8 CIM Elements

12.8.1 General

Table 127 describes the CIM elements for FabricVirtualFabrics.

Table 127 - CIM Elements for FabricVirtualFabrics

Element Name	Requirement	Description
12.8.2 CIM_Component (AdminDomain to Partitioning CS)	Mandatory	Associates the partitioning ComputerSystems representing the underlying physical switches to the AdminDomain representing the SAN.
12.8.3 CIM_ElementConformsToProfile (SAN AdminDomain to Virtual Fabrics RegisteredProfile)	Mandatory	Ties the SAN AdminDomain to the registered profile for Virtual Fabrics.

12.8.2 CIM_Component (AdminDomain to Partitioning CS)

Associates the partitioning ComputerSystems representing the underlying physical switches to the AdminDomain representing the SAN. This gives the client another means to determine which ComputerSystems are part of the Fabric versus those that are part of the underlying

physical topology. ComputerSystems representing Fibre Channel switches are associated to the AdminDomain representing the fabric (see clause 5 the Fabric Profile).

Created By: External

Modified By: Static

Deleted By: External

Requirement: Mandatory.

Table 128 describes class CIM_Component (AdminDomain to Partitioning CS).

**Table 128 - SMI Referenced Properties/Methods for CIM_Component
(AdminDomain to Partitioning CS)**

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	The reference to the AdminDomain representing the SAN (OtherIdentifyingInfo contains 'SAN' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions').
PartComponent		Mandatory	The reference to the partitioning ComputerSystem.

12.8.3 CIM_ElementConformsToProfile (SAN AdminDomain to Virtual Fabrics RegisteredProfile)

The CIM_ElementConformsToProfile ties SAN AdminDomain to the registered profile for Virtual Fabrics.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 129 describes class CIM_ElementConformsToProfile (SAN AdminDomain to Virtual Fabrics RegisteredProfile).

**Table 129 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile
(SAN AdminDomain to Virtual Fabrics RegisteredProfile)**

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	A SAN AdminDomain (OtherIdentifyingInfo contains 'SAN' with a corresponding 'SNIA:DetailedType' in IdentifyingDescriptions') that represents the Virtual Fabric.
ConformantStandard		Mandatory	RegisteredProfile instance describing the Virtual Fabrics profile.

EXPERIMENTAL

STABLE

13 Switch Profile

13.1 Description

13.1.1 General

The Switch Profile models logical and physical aspects of a Fibre Channel switch. The ComputerSystem class constitutes the center of the switch model (and is the top level object which the profile registration points to). An instance of a ComputerSystem is identified as a switch by the property Dedicated set to "switch".

This profile includes discovery components including ports, port statistics, product information, software, and chassis information. It also includes configuration of the switch including switch and port state change, port speed, switch name, symbolic names, and DomainID.

Both the Switch and Port have a capabilities class, FCSwitchCapabilities and FCPortCapabilities, respectively, defining which configuration options are supported by the switch. The capabilities define what components are configurable and any restrictions that apply. Except for state change, an associated settings class is defined for both the switch and port, FCSwitchSettings and FCPortSettings, which the client uses to request configuration changes to the Switch or Port, respectively. A setting does not necessarily result in a change to the underlying Switch or Port. The client can determine whether the setting was applied by looking at the associated property in the Switch or Port class.

The model for configuration is made up of three components, capabilities, settings, and the ManagedElements, ComputerSystem and FCPort, as shown in Figure 20. The capabilities define what components are configurable and any restrictions that apply. The settings define what the client requests, and the ManagedElements expose the actual changes that were applied.

The ComputerSystem (Dedicated as Switch) and FCPort classes have the method RequestStateChange() for requesting that the state be changed and an associated property RequestedState on the classes which indicates the current state change that has been requested. The property EnabledState can be examined to determine whether the device has completed the state change.

If a switch is modular, for instance if the switch is comprised of multiple blades on a backplane, LogicalModule can optionally be used to model each sub-module, and as an aggregation point for the switch ports. This is described in the Blade Subprofile.

FCPort describes the logical aspects of the port link and the data layers. PhysicalConnector models the physical aspects of a port. An instance of the FCPortStatistics class is expected for each instance of the FCPort class. FCPortStatistics expose real time port health and traffic information.

If the instrumentation is embedded in a switch, it shall provide a switch profile implementation for the hosting switch, and it may proxy a switch profile implementation for other switches reported in the Fabric Profile.

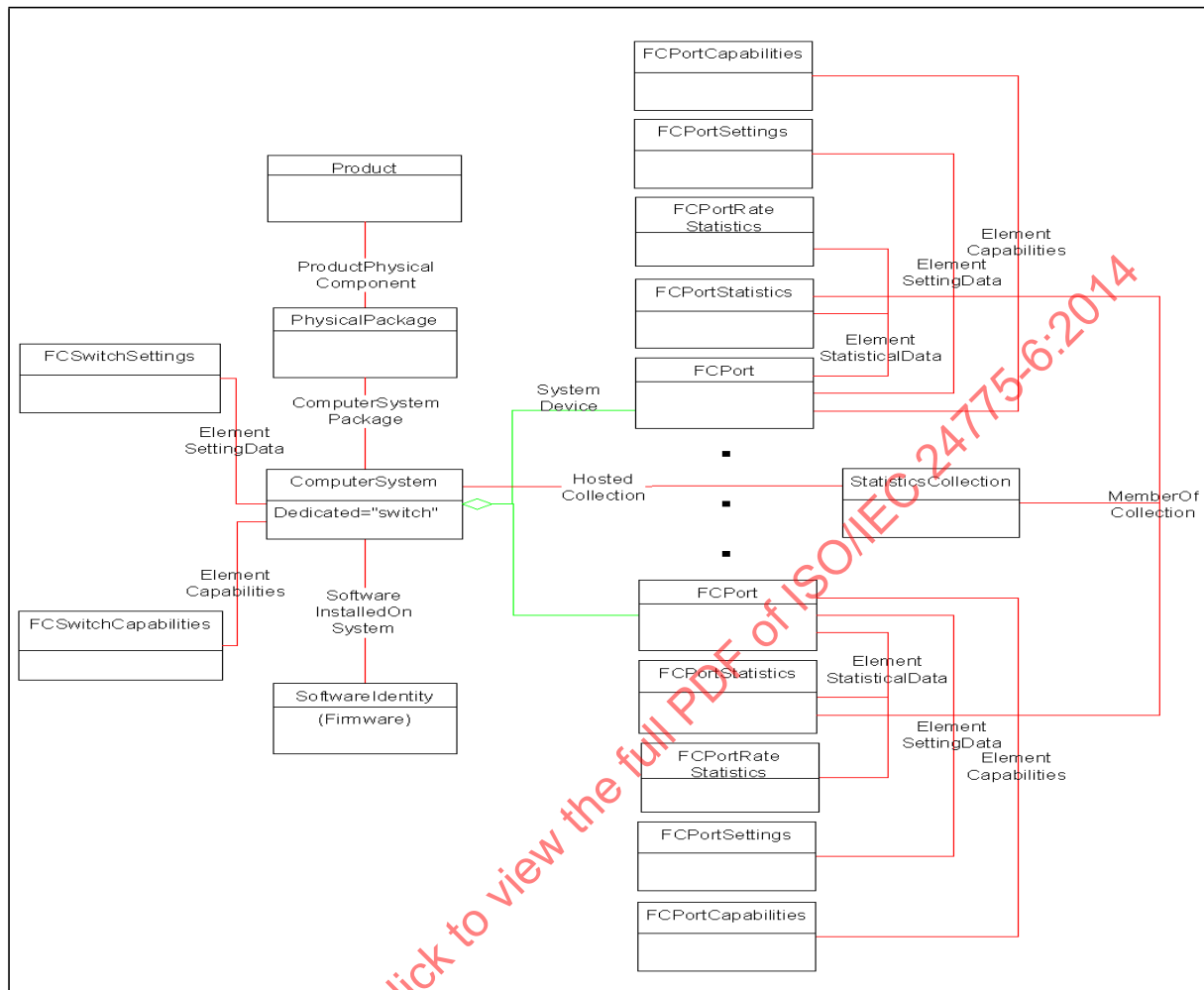


Figure 20 - Switch Instance Diagram

13.1.2 FC Port Settings and Capabilities

Capabilities describe the possible features that a ManagedElement supports. Settings are used to describe the requested configuration. The ManagedElement itself describes what settings have been applied and operating.

- For FC Port Type, there are settings that are not in the actual ManagedElement, FCPort.Types. These are settings that allow a subrange of possible port types. They are:
- a G_Port is a Switch Port that is capable of either operating as an E_Port or F_Port. A G_Port determines through Port Initialization whether it operates as an E_Port or as an F_Port;
- a GL_Port is a G_Port that is also capable of operating as an FL_Port;
- a Fx_Port is a switch port capable of operating as an F_Port or FL_Port.

The actual FCPort when operating shall run one of the port types as per FC-GS-6. In most cases a switch has a default setting to autonegotiate, which in most cases equates to GL or G being set in FCPortSetting.RequestedType. It is required that this setting, FCPortSetting.RequestedType, be shown regardless of whether it was set administratively or is the default behavior of the switch. FCPortSetting.RequestedType represents a setting that the administrator can understand and clearly identify why a switch port ends up running a particular port type. If the switch does not support setting the port type, the

RequestedTypesSupported array will be empty. It is valid to have a port type of "Unknown" until the link has been established. In 13.8.15 CIM_FCPortSettings, the table is showing the relationship of particular port types and how the capabilities relate to the setting for these port types.

The same concept applies for FCPort settings for speed except there is a separate property indicating auto negotiate, FCPortSettings.AutoSenseSpeed (LogicalPortSettings.AutoSenseSpeed). Note that this setting may have been previously set through some other administrative interface (e.g., CLI), but should still be reported in FCPortSettings.RequestedSpeed. If FCPortSetting.AutoSenseSpeed is true, then the value of FCPortSettings.RequestedSpeed is ignored and the speed will be negotiated by the hardware. If it is disabled, the port will operate at the speed configured in FCPortSettings.RequestedSpeed.

FCPortSettings.RequestedSpeed allows the port speed to be administratively set (WRITE qualifier). It also indicates to the client that the port has been administratively set (now or at a previous time). This property can only be set administratively if FCPortCapabilities.RequestedSpeedsSupported[] is not empty, and may only be set to one of the values in FCPortCapabilities.RequestedSpeedsSupported[].

FCPortCapabilities.RequestedSpeedsSupported indicates whether the device allows the speed to be administratively set. For instance, a 4 GB/s port may allow 1, 2, and 4 GB/s. FCPort.Speed (LogicalPort.Speed) represents the actual speed the port is running and a speed of zero represents that the link has not been established.

13.1.3 Trunking

Trunking describes from a switch perspective which ports are working together passing frames using the class RedundancySet. The RedundancySet has a property TypeOfSet which is used to identify what type of redundancy or trunking is occurring among the switch ports associated to the RedundancySet using MemberOfCollection. Figure 21 shows a Trunking instance.

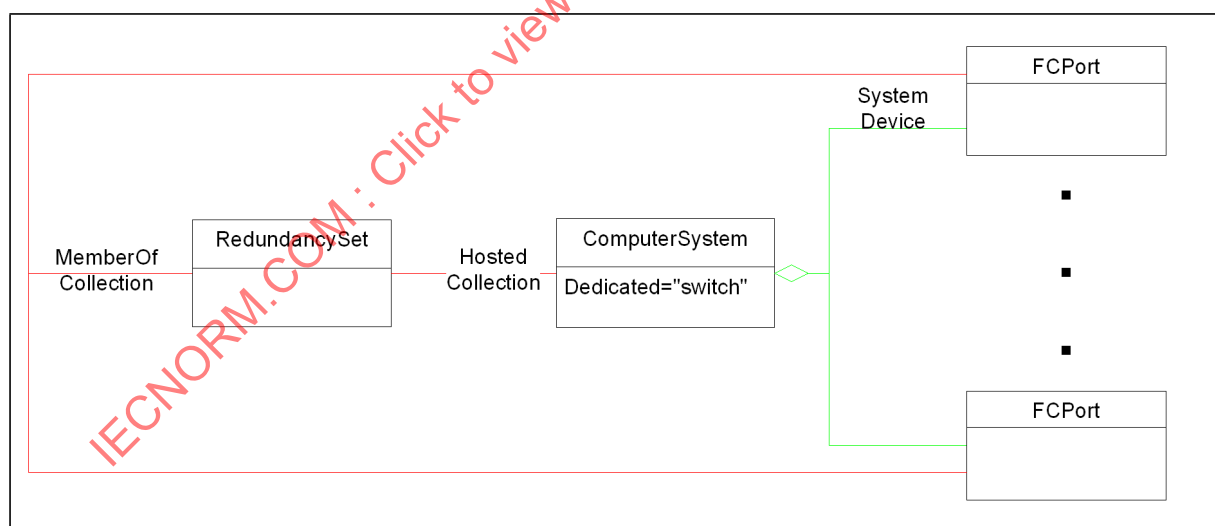


Figure 21 - Trunking Instance Diagram

EXPERIMENTAL

13.1.4 DetailedPortState, PortAvailability, OperationalStatus, and EnabledState

Table 2 - "Port OperationalStatus" is the starting point for the status of FCPort. It is a mandatory property of clause 5 Fabric Profile. EnabledState is also mandatory in clause 5: Fabric Profile, but only as the value Not Applicable. EnabledState has additional values when Clause 13 Switch Profile is implemented. Those values are shown in Table 130 which also shows how EnabledState relates to OperationalStatus. Table 130 and Table 131 contain

specifics about the expected values of the properties for FCPorts DetailedPortState and PortAvailability and their relationship to OperationalStatus and EnabledState.

In Table 130 are specifics about the expected values of the properties for FCPorts DetailedPortState and its relationship to OperationalStatus, EnabledState and PortAvailability.

Table 130 - DetailedPortState for FCPort

DetailedPortState	OperationalStatus	EnabledState	PortAvailability
Transceiver Validation	Starting	Starting	Available
Locking To Reference Signal	Starting	Starting	Available
Synchronized	OK	Enabled	Available
Invalid Transceiver	Error	Enabled, but offline	Available
No Light ^a	OK	Enabled, but offline	Available
Cannot Synchronize	Error	Disabled	Available
Laser Fault	Error	Disabled	Available
Port Fault	Error	Disabled	Available
Diagnostic Fault	Error	Disabled	Available
Unknown	Unknown	Disabled	Any value in Table 131

^a When there is no light, this only indicates that the transceiver is not receiving light and the transceiver is operational. Typically this case occurs when there is no cable attached, but can also be caused by a broken/defective cable.

In Table 131 are specifics about the expected values of the properties for PortAvailability and its relationship to OperationalStatus, EnabledState and DetailedPortState.

Table 131 - PortAvailability for FCPort

PortAvailability	OperationalStatus	EnabledState	DetailedPortState
Available	Any value in Table 2, "Port OperationalStatus,"	Any value in Table 130 except Unknown	Any value in Table 130
Not Installed	Stopped	Disabled	Unknown
No Transceiver	Stopped	Disabled	Unknown
Not Licensed	Stopped	Disabled	Unknown

EXPERIMENTAL

EXPERIMENTAL

13.1.5 Conditional Classes and Properties

Table 133 contains classes that are conditional on the implementation of other profiles.

- CIM_ComputerSystem (Partitioned Switch) is conditional on the implementation of the Switch Partitioning Profile.
- CIM_ElementSettingData (FCSwitchSettings to ComputerSystem) is conditional on the implementation of the Switch Configuration Data Profile.
- CIM_FCPort (Partitioned Switch FCPort) is conditional on the implementation of the Switch Partitioning Profile.

In addition, some properties of the Switch classes are conditional on implementation of other profiles.

- The ElementName, OtherIdentifyingInfo, IdentifyingDescriptions and EnabledState of CIM_ComputerSystem (Switch) are required if the Fabric Profile is implemented.
- EnabledState of CIM_FCPort (Switch FCPort) is required if the Fabric Profile is implemented.
- PortDiscriminator of CIM_FCPort (Switch FCPort) is required if the Switch Partitioning Profile is implemented.

EXPERIMENTAL

EXPERIMENTAL

13.1.6 Switch Predefined Indications

If the optional Experimental Indication profile is supported by an implementation of the Switch profile, it shall implement the SNIA_IndicationConfigurationService and its associated SNIA_IndicationConfigurationCapabilities. If the implementation supports predefined IndicationFilters or predefined IndicationFilterCollections this shall be indicated in the SupportedFeatures property of the SNIA_IndicationConfigurationCapabilities. If a value "3" is present, it means the implementation supports predefined IndicationFilters. If a value of "5" is present, it means the implementation supports predefined IndicationFilterCollections.

Figure 22 illustrates classes that shall be populated by the Switch profile if both "3" and "5" are present in the SupportedFeatures property.

IECNORM.COM : Click to view the full PDF of ISO/IEC 24775-6:2014

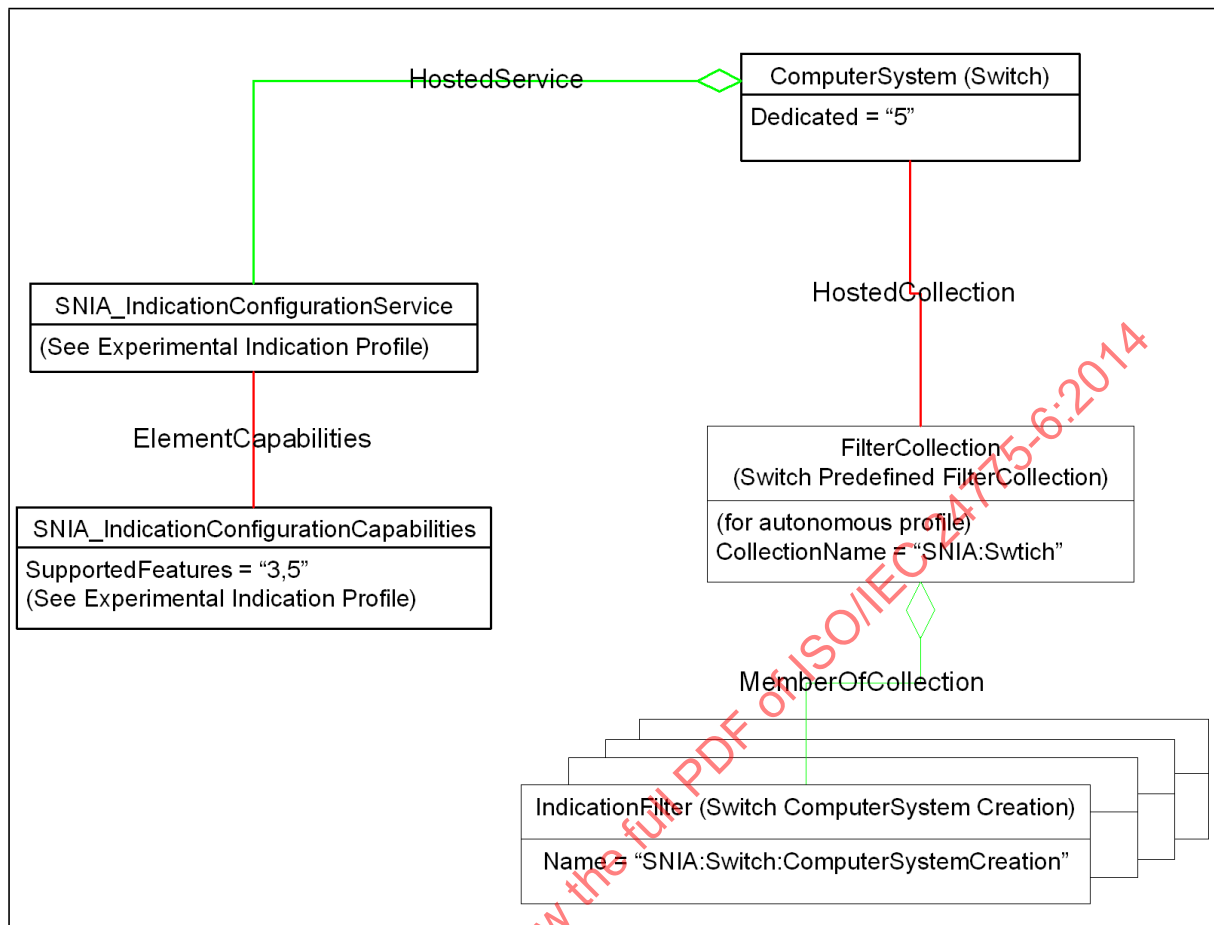


Figure 22 - Predefined Switch IndicationFilters and Collection

The SNIA_IndicationConfigurationService is hosted on the switch ComputerSystem. The FilterCollection for switch is also hosted on the switch ComputerSystem. The switch FilterCollection has members which are all the predefined IndicationFilters supported by the implementation. This shall include all Mandatory IndicationFilters of the Switch Profile. But it should also include any IndicationFilter that has been predefined by the implementation. This may include conditional, optional or vendor specific IndicationFilters supported by the implementation.

The switch FilterCollection shall have the CollectionName "SNIA:Switch". Each of the predefined filters shall have the Name property as defined for the IndicationFilter. In Figure 22 the name of the IndicationFilter (Switch ComputerSystem Creation) is "SNIA:Switch:ComputerSystemCreation". For vendor specific IndicationFilters (not defined in this International Standard), the Name of the filter would be of the form ORG_ID":Switch:"UNIQUE_ID, where ORGID is the designation of the vendor that is providing the implementation.

EXPERIMENTAL

13.2 Health and Fault Management

The following classes report possible Health and Fault information through LifeCycle indications:

- ComputerSystem
- FCPort

These LifeCycle indications are more fully described in ISO/IEC 24775-3, *Information Technology – Storage Management, Part 3 Common Profiles*.

Also in Table 133, “CIM Elements for Switch” is a list of AlertIndications which may also be indicators for Health and Fault Management.

13.3 Cascading Considerations

None.

13.4 Dependencies on Profiles, Subprofiles, and Packages

Table 132 describes the supported profiles for Switch.

Table 132 - Supported Profiles for Switch

Profile Name	Organization	Version	Requirement	Description
Blades	SNIA	1.5.0	Optional	
Access Points	SNIA	1.3.0	Optional	
Switch Configuration Data	SNIA	1.1.0	Optional	
Physical Package	SNIA	1.5.0	Mandatory	
Software	SNIA	1.4.0	Mandatory	
FabricSwitchPartitioning	SNIA	1.5.0	Optional	
Power Supply	SNIA	1.0.1	Optional	
Fan	SNIA	1.0.1	Optional	
Record Log	DMTF	1.0.1	Optional	
Indication	SNIA	1.5.0	Mandatory	
Experimental Indication	SNIA	1.5.0	Optional	Experimental.
Launch In Context	DMTF	1.0.0	Optional	Experimental. See DSP1102, version 1.0.0

13.5 Methods of this Profile

None in this International Standard.

13.6 Client Considerations and Recipes

13.6.1 Enable FCPort

```
// DESCRIPTION
// This recipe describes how to enable a port on a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to be enabled is known as $Port.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
```

```
false,
    {"RequestedStatesSupported"})
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {
    <ERROR! The required port capabilities are not available>
}

// Step 2. Verify that the port can be enabled.
if (!contains(2, $PortCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Enabling the specified port is not supported>
}

// Step 3. Verify that the port is in a state in which enabling is
// appropriate.
if ($Port.EnabledState != 2 && $Port.RequestedState == 5) {

    // Step 4. Enable the port.
    %InArguments["RequestedState"] = 2// "Enabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Port.GetObjectPath(),
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Port successfully enabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not
        Supported"
        %InArguments["RequestedState"] = 2// "Enabled"
        %InArguments["TimeoutPeriod"] = 0// No timeout
        #ReturnValue = InvokeMethod($Port.GetObjectPath(),
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) {// "Completed with No Error"
            <EXIT! Port successfully enabled>
        } else {
            <ERROR! Port state transition failed>
        }
    }
} else {
    <ERROR! The specified port is already enabled or currently in a
    state transition>
}
```

13.6.2 Disable Port

```
// DESCRIPTION
// This recipe describes how to disable a port on a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to be disabled is known as $Port.
```

```

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {
    <ERROR! The required port capabilities are not available>
}

// Step 2. Verify that the port can be disabled.
if (!contains(3, $Capabilities.RequestedStatesSupported)) {
    <EXIT! Disabling the specified port is not supported>
}

// Step 3. Verify that the port is in a state in which disabling is
// appropriate.
if ($Port.EnabledState != 3 && $Port.RequestedState == 5) {

    // Step 4. Disable the port.
    %InArguments["RequestedState"] = 3// "Disabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Port.getObjectPath(),
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Port successfully disabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not
        Supported"
        %InArguments["RequestedState"] = 3 // "Disabled"
        %InArguments["TimeoutPeriod"] = 0 // No timeout
        #ReturnValue = InvokeMethod($Port.getObjectPath(),
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) {// "Completed with No Error"
            <EXIT! Port successfully disabled>
        } else {
            <ERROR! Port state transition failed>
        }
    }
} else {
    <ERROR! The specified port is already disabled or currently in a
    state transition>
}

```

}

13.6.3 Enable Switch

```
// DESCRIPTION
// This recipe describes how to enable a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to enable is known and defined in the
//    variable $Switch->.

// MAIN
// Step 1. Retrieve the relevant Switch instance information.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"EnabledState", "RequestedState"})

// Step 2. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 3. Verify that the Switch can be enabled.
if (!contains(2, $SwitchCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Enabling the specified Switch is not supported>
}

// Step 4. Verify that the Switch is in a state in which enabling is
// appropriate.
if ($Switch.EnabledState != 2 && $Switch.RequestedState == 5) {

    // Step 5. Enable the Switch.
    %InArguments["RequestedState"] = 2// "Enabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Switch->,
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Switch successfully enabled>
    }
}
```

```

    } else if (#ReturnValue == 4098) { // "Timeout Parameter Not
        Supported"
        %InArguments["RequestedState"] = 2 // "Enabled"
        %InArguments["TimeoutPeriod"] = 0 // No timeout
        #ReturnValue = InvokeMethod($Switch->,
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) { // "Completed with No Error"
            <EXIT! Switch successfully enabled>
        } else {
            <ERROR! Switch state transition failed>
        }
    }
} else {
    <ERROR! The specified Switch is already enabled or currently in
        a state transition>
}

```

13.6.4 Disable Switch

```

// DESCRIPTION
// This recipe describes how to disable a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to disable is known and defined in the
//     variable $Switch->.

// MAIN
// Step 1. Retrieve the relevant Switch instance information.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"EnabledState", "RequestedState"})

// Step 2. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 3. Verify that the Switch can be disabled.
if (contains(3, $SwitchCapabilities[0].RequestedStatesSupported)) {

```

```
<EXIT! Disabling the specified Switch is not supported>
}

// Step 4. Verify that the Switch is in a state in which disabling is
// appropriate.
if ($Switch.EnabledState != 3 && $Switch.RequestedState == 5) {

    // Step 5. Disable the Switch.
    %InArguments["RequestedState"] = 3// "Disabled"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Switch->,
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) {// "Completed with No Error"
        <EXIT! Switch successfully disabled>
    } else if (#ReturnValue == 4098) {// "Timeout Parameter Not
        Supported"
        %InArguments["RequestedState"] = 3// "Disabled"
        %InArguments["TimeoutPeriod"] = 0// No timeout
        #ReturnValue = InvokeMethod($Switch->,
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) {// "Completed with No Error"
            <EXIT! Switch successfully disabled>
        } else {
            <ERROR! Switch state transition failed>
        }
    } else {
        <ERROR! Switch state transition failed>
    }
} else {
    <ERROR! The specified Switch is already disabled or currently in
    a state transition>
}
```

13.6.5 Reset Switch

```
// DESCRIPTION
// This recipe describes how to reset a Fibre Channel Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to reset is known and defined in the
//     variable $Switch->.
//
// MAIN
// Step 1. Retrieve the relevant Switch instance information.
$Switch = GetInstance($Switch->,
    false,
```

```

        false,
        false,
        {"EnabledState", "RequestedState"})

// Step 2. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedStatesSupported"})
if ($SwitchCapabilities[] == null || SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 3. Verify that the Switch can be reset.
if (contains(11, $SwitchCapabilities[0].RequestedStatesSupported)) {
    <EXIT! Resetting the specified Switch is not supported>
}

// Step 4. Verify that the Switch is in a state in which resetting is
// appropriate.
if ($Switch.EnabledState == 2 && $Switch.RequestedState == 5) {

    // Step 5. Reset the Switch.
    %InArguments["RequestedState"] = 11// "Reset"
    // Timeout request after 90 seconds
    %InArguments["TimeoutPeriod"] = 00000000000130.000000:000
    #ReturnValue = InvokeMethod($Switch->,
        "RequestStateChange",
        %InArguments,
        %OutArguments)
    if (#ReturnValue == 0) { // "Completed with No Error"
        <EXIT! Switch successfully reset>
    } else if (#ReturnValue == 4098) { // "Timeout Parameter Not
        Supported"
        %InArguments["RequestedState"] = 11// "Reset"
        %InArguments["TimeoutPeriod"] = 0// No timeout
        #ReturnValue = InvokeMethod($Switch->,
            "RequestStateChange",
            %InArguments,
            %OutArguments)
        if (#ReturnValue == 0) { // "Completed with No Error"
            <EXIT! Switch successfully reset>
        } else {
            <ERROR! Switch state transition failed>
        }
    } else {
        <ERROR! Switch state transition failed>
    }
}

```

```
    }  
  } else {  
    <ERROR! The specified Switch is already reset or currently in  
      a state transition>  
  }  
}
```

13.6.6 Set Port Speed

```
// DESCRIPTION  
// This recipe describes how to modify the speed of a port on a Fibre  
// Channel  
  
// Switch.  
//  
// PREEXISTING CONDITIONS AND ASSUMPTIONS  
// 1. The instance of the port to whose speed to modify is known as  
//    $Port.  
// 2. The desired port speed is known and defined in the variable #Speed.  
  
// MAIN  
// Step 1. Retrieve the capabilities of the port.  
$PortCapabilities[] = Associators($Port.getObjectPath(),  
  "CIM_ElementCapabilities",  
  "CIM_FCPortCapabilities",  
  "ManagedElement",  
  "Capabilities",  
  false,  
  false,  
  {"AutoSenseSpeedConfigurable", "RequestedSpeedsSupported"})  
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {  
  <ERROR! The required port capabilities are not available>  
}  
$Capabilities = $PortCapabilities[0]  
  
// Step 2. Verify that the port speed can be set to the specified speed.  
if (contains(#Speed, $Capabilities.RequestedSpeedsSupported)) {  
  
  // Step 3. Retrieve the port settings.  
  $Settings[] = Associators($Port.getObjectPath(),  
    "CIM_ElementSettingData",  
    "CIM_FCPortSettings",  
    "ManagedSetting",  
    "SettingData",  
    false,  
    false,  
    {"InstanceID", "AutoSenseSpeed", "RequestedSpeed"})  
  if ($Settings[] == null || Settings[].length != 1) {  
    <ERROR! The required port settings are not available>  
  }  
  $PortSetting = $Settings[0]  
  
  // Step 4. Port speed is ignored unless AutoSenseSpeed is disabled,
```

```

    if ($PortSetting.AutoSenseSpeed) {
        if ($Capabilities.AutoSenseSpeedConfigurable) {
            $PortSetting.AutoSenseSpeed = false
        } else {
            //Unlikely, but not an error
        }
    }
}

// Step 5. Modify the port speed to the specified speed.
$PortSetting.RequestedSpeed = #Speed
ModifyInstance($PortSetting.GetObjectPath(),
    $PortSetting,
    false,
    {"AutoSenseSpeed", "RequestedSpeed"})

// Step 6. Verify that the port speed modification was applied.
$Port = GetInstance($Port.GetObjectPath(),
    false,
    false,
    false,
    {"Speed"})
if ($Port.Speed == #Speed) {
    <EXIT! Port speed modified successfully>
} else {
    <ERROR! Port speed was not modified as specified>
}
} else {
    <EXIT! Specified port speed is not supported>
}

```

13.6.7 Set Port Type

```

// DESCRIPTION
// This recipe describes how to modify the port type on a Fibre Channel
// Switch.

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to whose type to modify is known as $Port.
// 2. The desired port type is known and defined in the variable #Type.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.GetObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"RequestedTypesSupported"})

```

```
if ($PortCapabilities[] == null || PortCapabilities[].length != 1) {
    <ERROR! The required port capabilities are not available>
}

// Step 2. Verify that the port type can be modified as specified.
$Capabilities = $PortCapabilities[0]
if (contains(#Type, $Capabilities.RequestedTypesSupported)) {

    // Step 3. Retrieve the port settings.
    $Settings[] = Associators($Port.GetObjectPath(),
        "CIM_ElementSettingData",
        "CIM_FCPortSettings",
        "ManagedSetting",
        "SettingData",
        false,
        false,
        {"RequestedType"})
    if ($Settings[] == null || Settings[].length != 1) {
        <ERROR! The required port settings are not available>
    }
    $PortSetting = $Settings[0]

    // Step 4. Modify the port type to the specified type.
    $PortSetting.RequestedType = #Type
    ModifyInstance($PortSetting.GetObjectPath(),
        $PortSetting,
        false,
        {"RequestedType"})

    // Step 5. Verify that the port type modification was applied.
    $Port = GetInstance($PortSetting.GetObjectPath(),
        false,
        false,
        false,
        {"RequestedType"})
    if ($PortSetting.RequestedType == #Type) {
        <EXIT! Port type request successfully>
    }
    <ERROR! Port type request was not modified as specified>
} else {
    <ERROR! Port type request cannot be set to specified type>
}
```

13.6.8 Set Fibre Channel Switch Principal Priority

```
// DESCRIPTION
// This recipe describes how to modify the Principal Priority of a Fibre
// Channel
// Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
```

```

//
// 1. A reference to the Switch whose Principal Priority to modify is
//      known and
//      defined in the variable $Switch->
// 2. The desired Principal Priority of the Switch is known as #Priority.

// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"PrincipalPrioritiesSupported"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 2. Verify that the Switch Principal Priority can be modified.
$Capabilities = $SwitchCapabilities[0]
if (!contains(5, $Capabilities.PrincipalPrioritiesSupported[])) {

    $SwitchSettings[] = Associators($Switch->,
        "CIM_ElementSettingData",
        "CIM_FCSwitchSettings",
        "ManagedElement",
        "SettingData",
        false,
        false,
        {"PrincipalPriority"})
    if ($SwitchSettings[] == null || $SwitchSettings[].length != 1) {
        <ERROR! Required Switch settings are not available>
    }
    $Settings = $SwitchSettings[0]

    // Step 3. Ensure a new Principal Priority is being set.
    if (#Priority != $Settings.PrincipalPriority)) {

        // Step 4. Modify the Principal Priority of the Switch.
        $Settings.PrincipalPriority = #Priority
        ModifyInstance($Settings.GetObjectPath(),
            $Settings,
            false,
            {"PrincipalPriority"})

        // Step 5. Verify that the Switch priority modification was applied.
        $Settings = GetInstance($Settings.GetObjectPath(),
            false,
            false,
            false,

```

```
        {"PrincipalPriority"})
    if ($Settings.PrincipalPriority == #Priority) {
        <EXIT! Switch Principal Priority was modified successfully>
    }
    <EXIT! Switch Principal Priority was not modified successfully>
} else {
    <ERROR! Principal Priority specified is already set>
}
} else {
    // "Not Applicable"
    <EXIT! The Switch does not support Principal Priority modification>
}
```

13.6.9 Set Switch Name

```
// DESCRIPTION
// This recipe describes how to modify the name of a Fibre Channel
// Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
//
// 1. A reference to the Switch whose name to modify is known and defined
//    in
//    the variable $Switch->
// 2. The desired name of the Switch is known as #Name.
//
// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"ElementNameEditSupported", "MaxElementNameLen"})
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {
    <ERROR! The required Switch capabilities are not available>
}

// Step 2. Verify that the Switch name can be modified.
$Capabilities = $SwitchCapabilities[0]
if ($Capabilities.ElementNameEditSupported) {

    // Step 3. Verify that the new name to be specified is within the
    // constraints of the name length supported by the Switch.
    if (#Name.length() < $Capabilities.MaxElementNameLen) {

        // Step 4. Retrieve the instance representing the Switch.
        $Switch = GetInstance($Switch->,
            false,
```

```

        false,
        false,
        {"ElementName"})

// Step 5. Modify the name of the Switch.
$Switch.ElementName = #Name
ModifyInstance($Switch->,
    $Switch,
    false,
    {"ElementName"})

// Step 6. Verify that the Switch name change was applied.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"ElementName"})
if (compare(#Name, $Switch.ElementName)) {
    <EXIT! Switch name was modified successfully>
}
<ERROR! Switch name was not modified successfully>
}
<ERROR! Specified Switch name exceeds length limit>
} else {
    <EXIT! The Switch does not support name modification>
}

```

13.6.10 Set Port Name

```

// DESCRIPTION
// This recipe describes how to modify the name of a Port on a Fibre
// Channel
// Switch
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. The instance of the port to whose type to modify is known as $Port.
// 2. The desired name of the port is known as #Name.

// MAIN
// Step 1. Retrieve the capabilities of the port.
$PortCapabilities[] = Associators($Port.getObjectPath(),
    "CIM_ElementCapabilities",
    "CIM_FCPortCapabilities",
    "ManagedElement",
    "Capabilities",
    false,
    false,
    {"ElementNameEditSupported", "MaxElementNameLen"})
if ($PortCapabilities[] == null || $PortCapabilities[].length != 1) {
    <ERROR! The required Port capabilities are not available>
}

```

```
// Step 2. Verify that the port name can be modified.
$Capabilities = $PortCapabilities[0]
if ($Capabilities.ElementNameEditSupported) {

    // Step 3. Verify that the new name to be specified is within the
    // constraints of the name length supported by the port.
    if (#Name.length() < $Capabilities.MaxElementNameLen) {

        // Step 4. Modify the name of the port.
        $Port.ElementName = #Name
        ModifyInstance($Port.getObjectPath(),
            $Port,
            false,
            {"ElementName"})

        // Step 5. Verify that the port name change was applied.
        $Port = GetInstance($Port.getObjectPath(),
            false,
            false,
            false,
            {"ElementName"})
        if (compare(#Name, $Port.ElementName)) {
            <EXIT! Port name was modified successfully>
        }
        <ERROR! Port name was not modified successfully>
    }
    <ERROR! Specified Port name exceeds length limit>
} else {
    <EXIT! The Port does not support name modification>
}
```

13.6.11 Set Fibre Channel Switch Preferred Domain ID

```
// DESCRIPTION
//
// This recipe describes how to modify the preferred Domain ID of a
// Switch.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch to reset is known and defined in the
// variable $Switch->.
// 2. The new preferred Domain ID to be set on the Switch is known as
// #DomainID.
//
// MAIN
// Step 1. Retrieve the capabilities of the Switch.
$SwitchCapabilities[] = Associators($Switch->,
    "CIM_ElementCapabilities",
    "CIM_FCSwitchCapabilities",
    "ManagedElement",
```

```

        "Capabilities",
        false,
        false,
        {"DomainIDConfigureable", "MinDomainID", "MaxDomainID"})
    if ($SwitchCapabilities[] == null || SwitchCapabilities[].length != 1) {
        <ERROR! The required Switch capabilities are not available>
    }

    // Step 2. Verify that the Switch's preferred Domain ID can be modified.
    $Capabilities = $SwitchCapabilities[0]
    if ($Capabilities.DomainIDConfigureable) {

        // Step 3. Verify that the desired Domain ID is within the permissible
        // range.
        if (#DomainID >= $Capabilities.MinDomainID
            && #DomainID <= $Capabilities.MaxDomainID) {

            // Step 4. Retrieve the Switch settings.
            $Settings[] = Associators($Switch->,
                "CIM_ElementSettingData",
                "CIM_FCSwitchSettings",
                "ManagedSetting",
                "SettingData",
                false,
                false,
                {"PreferredDomainID"})
            if ($Settings[] == null || Settings[].length != 1) {
                <ERROR! The required Switch settings are not available>
            }
            $SwitchSetting = $Settings[0]

            // Step 5. Modify the Switch Domain ID to the specified preferred
            // value.
            $SwitchSetting.PreferredDomainID = #DomainID
            ModifyInstance($SwitchSetting.GetObjectPath(),
                $SwitchSetting,
                false,
                {"PreferredDomainID"})

            // Step 6. Verify that the Switch Domain ID modification was
            // applied.
            $Switch = GetInstance($Switch->,
                false,
                false,
                false,
                {"IdentifyingDescriptions", "OtherIdentifyingInfo"})
            // NOTE: The Domain ID value is contained in the
            // OtherIdentifyingInfo
            // property at the same index as the "DomainID" element index in the
            // IdentifyingDescriptions property.
            #index = -1
            while (#i < $Switch.IdentifyingDescriptions[].length

```

```
&& #index < 0) {  
    if ($Switch.IdentifyingDescriptions[#i] == "DomainID") {  
        #index = #i  
    }  
}  
if (#index >= 0 && $Switch.OtherIdentifyingInfo[#index] ==  
    #DomainID) {  
    <EXIT! Switch Domain ID successfully modified>  
}  
<ERROR! Switch Domain ID was not modified as specified>  
} else {  
    <ERROR! Domain ID specified is not within permitted range>  
}  
} else {  
    <EXIT! Domain ID configuration on the specified Switch is not  
        supported>  
}
```

13.6.12 Lock Fibre Channel Switch Domain ID

```
// DESCRIPTION  
//  
// This recipe describes how to set the Domain ID Lock of a Switch.  
//  
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS  
// 1. A reference to the Switch whose Domain ID to lock is known and  
//     defined  
// in the variable $Switch->.  
  
// MAIN  
// Step 1. Retrieve the capabilities of the Switch.  
$SwitchCapabilities[] = Associators($Switch->,  
    "CIM_ElementCapabilities",  
    "CIM_FCSwitchCapabilities",  
    "ManagedElement",  
    "Capabilities",  
    false,  
    false,  
    {"DomainIDLockedSupported"})  
if ($SwitchCapabilities[] == null || $SwitchCapabilities[].length != 1) {  
    <ERROR! The required Switch capabilities are not available>  
}  
  
// Step 2. Verify that the Switch's Domain ID Lock can be set.  
$Capabilities = $SwitchCapabilities[0]  
if ($Capabilities.DomainIDLockedSupported) {  
  
    // Step 3. Retrieve the Switch settings.  
    $Settings[] = Associators($Switch->,  
        "CIM_ElementSettingData",  
        "CIM_FCSwitchSettings",  
        "ManagedSetting",
```

```

        "SettingData",
        false,
        false,
        {"DomainIDLocked", "PreferredDomainID"})
if ($Settings[] == null || Settings[].length != 1) {
    <ERROR! The required Switch settings are not available>
}
$SwitchSetting = $Settings[0]
#PreferredDomainID = $SwitchSetting.PreferredDomainID

// Step 4. Verify that the Domain ID is not already locked.
if ($SwitchSetting.DomainIDLocked) {
    <EXIT! The Domain ID Lock is already set>
}

// Step 5. Lock the Switch Domain ID.
$SwitchSetting.DomainIDLocked = true
ModifyInstance($SwitchSetting.GetObjectPath(),
    $SwitchSetting,
    false,
    {"DomainIDLocked"})

// Step 6. Verify that the Switch Domain ID specifies the preferred
// Domain ID.
$Switch = GetInstance($Switch->,
    false,
    false,
    false,
    {"IdentifyingDescriptions", "OtherIdentifyingInfo"})
// NOTE: The Domain ID value is contained in the OtherIdentifyingInfo
// property at the same index as the "DomainID" element index in the
// IdentifyingDescriptions property.
#index = -1
while (#i < $Switch.IdentifyingDescriptions[].length && #index < 0) {
    if ($Switch.IdentifyingDescriptions[#i] == "DomainID") {
        #index = #i
    }
}
if (#index >= 0 &&
    $Switch.OtherIdentifyingInfo[#index] == #PreferredDomainID) {
    <EXIT! Switch Domain ID successfully locked>
}
<ERROR! Switch Domain ID does not reflect the preferred Domain ID>
} else {
    <EXIT! Domain ID configuration on the specified Switch is not
        supported>
}

```

13.7 Registered Name and Version

Switch version 1.5.0 (Autonomous Profile).

13.8 CIM Elements

13.8.1 General

Table 133 describes the CIM elements for Switch.

Table 133 - CIM Elements for Switch

Element Name	Requirement	Description
13.8.2 CIM_ComputerSystem (Partitioned Switch)	Conditional	Conditional requirement: Support for the Switch Partitioning profile. Represents the Partitioned Switch.
13.8.3 CIM_ComputerSystem (Switch)	Mandatory	Represents the Switch Associated to RegisteredProfile.
13.8.4 CIM_ComputerSystemPackage	Mandatory	Associates PhysicalPackage to the ComputerSystem (Switch).
13.8.5 CIM_ElementCapabilities (FCPort to FCPortCapabilities)	Mandatory	Associates FCPort to the FCPortCapabilities.
13.8.6 CIM_ElementCapabilities (System to FCSwitchCapabilities)	Mandatory	Associates FCSwitchCapabilities to the ComputerSystem (Switch).
13.8.7 CIM_ElementSettingData (FCPortSettings to FCPort)	Optional	Associates FCPortSettings to FCPort.
13.8.8 CIM_ElementSettingData (FCSwitchSettings to ComputerSystem)	Conditional	Conditional requirement: Support for the Switch Configuration Data profile. Associates FCSwitchSettings to ComputerSystem (Switch).
13.8.9 CIM_ElementStatisticalData (FCPortRateStatistics to FCPort)	Optional	Associates FCPortRateStatistics to the FCPort.
13.8.10 CIM_ElementStatisticalData (FCPortStatistics to FCPort)	Mandatory	Associates FCPortStatistics to the FCPort.
13.8.11 CIM_FCPort (Partitioned Switch FCPort)	Conditional	Conditional requirement: Support for the Switch Partitioning profile. Fibre Channel Switch Port for a Virtual Switch.
13.8.12 CIM_FCPort (Switch FCPort)	Mandatory	Fibre Channel Switch Port.
13.8.13 CIM_FCPortCapabilities	Mandatory	Switch Port Capabilities.
13.8.14 CIM_FCPortRateStatistics	Optional	Fibre Channel Switch Port Rate Statistics.
13.8.15 CIM_FCPortSettings	Optional	Switch Port Settings.
13.8.16 CIM_FCPortStatistics	Mandatory	Fibre Channel Switch Port Statistics.
13.8.17 CIM_FCSwitchCapabilities	Mandatory	Fibre Channel Switch Capabilities.
13.8.18 CIM_FCSwitchSettings	Mandatory	Fibre Channel Switch Settings.
13.8.19 CIM_FilterCollection (Switch Predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This is a collection of predefined IndicationFilters to which a client may subscribe.
13.8.20 CIM_HostedCollection (Redundancy Set)	Optional	Associates the redundancy set of FCPorts to the ComputerSystem representing the switch.

Table 133 - CIM Elements for Switch

Element Name	Requirement	Description
13.8.21 CIM_HostedCollection (Statistics Collection)	Mandatory	Associates the Statistics Collection to the ComputerSystem representing the switch.
13.8.22 CIM_HostedCollection (Switch to predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).
13.8.23 CIM_IndicationFilter (Switch ComputerSystem Creation)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new Switch computer system instance.
13.8.24 CIM_IndicationFilter (Switch ComputerSystem Deletion)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a switch computer system instance.
13.8.25 CIM_IndicationFilter (Switch ComputerSystem OperationalStatus)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the modification of OperationalStatus of a switch computer system instance.
13.8.26 CIM_IndicationFilter (Switch FCPort OperationalStatus)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the modification of OperationalStatus of an FCPort instance in the Switch.
13.8.27 CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus)	Conditional	Deprecated. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a switch ComputerSystem instance.
13.8.28 CIM_IndicationFilter (WQL Switch FCPort OperationalStatus)	Conditional	Deprecated. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a switch FCPort instance.

Table 133 - CIM Elements for Switch

Element Name	Requirement	Description
13.8.29 CIM_MemberOfCollection (FCPort to RedundancySet)	Optional	Associates the FCPort to the RedundancySet.
13.8.30 CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection)	Mandatory	Associates the NetworkPortStatistics to the StatisticalCollection.
13.8.31 CIM_MemberOfCollection (Predefined Filter Collection to Switch Filters)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This associates the switch predefined FilterCollection to the predefined Filters supported by the switch.
13.8.32 CIM_ProtocolEndpoint	Optional	The endpoint of a link (ActiveConnection).
13.8.33 CIM_RedundancySet	Optional	The class RedundancySet along with the association MemberOfCollection in this profile is used to show port aggregation for Fibre Channel trunking.
13.8.34 CIM_StatisticsCollection	Mandatory	Collection to aggregate the FCPortStatistics for each switch.
13.8.35 CIM_SystemDevice	Mandatory	Associates FCPort to the ComputerSystem (Switch).
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	New Switch Instance. See 13.8.23 CIM_IndicationFilter (Switch ComputerSystem Creation).
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	Deletion of Switch Instance. See 13.8.24 CIM_IndicationFilter (Switch ComputerSystem Deletion)
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated. Deprecated WQL - Modification of OperationalStatus in Switch Instance. See 13.8.27 CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated. Deprecated WQL - Modification of OperationalStatus in FC Port Instance. See 13.8.28 CIM_IndicationFilter (WQL Switch FCPort OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus	Mandatory	CQL - Modification of OperationalStatus in Switch Instance. See 13.8.25 CIM_IndicationFilter (Switch ComputerSystem OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus	Mandatory	CQL - Modification of OperationalStatus in FC Port Instance. See 13.8.26 CIM_IndicationFilter (Switch FCPort OperationalStatus).

13.8.2 CIM_ComputerSystem (Partitioned Switch)

Represents the Partitioned Switch. A Partitioned Switch shall contain 'Virtual Switch' in one of its entries in OtherIdentifyingInfo with a corresponding value of 'SNIA:DetailedType' in the IdentifyingDescriptions array property.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Support for the Switch Partitioning profile.

Table 134 describes class CIM_ComputerSystem (Partitioned Switch).

Table 134 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned Switch)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	See the CreationClassName definition in 13.8.3 CIM_ComputerSystem (Switch).
Name	D	Mandatory	See the Name definition in 13.8.3 CIM_ComputerSystem (Switch).
ElementName		Conditional	Conditional requirement: Support for the Fabric profile. See the ElementName definition in 5.8.10 CIM_ComputerSystem (Switch). User-friendly name. Can be set if FCSwitchCapabilities.ElementNameEditSupported for the switch is True.
NameFormat		Mandatory	See the NameFormat definition in 13.8.3 CIM_ComputerSystem (Switch).
OtherIdentifyingInfo	C	Mandatory	DomainID stored in decimal format, with a value between 0 and 255. Another value corresponding to 'SNIA:DetailedType' shall contain 'Virtual Switch'. Other entries in OtherIdentifyingInfo may also exist.
OperationalStatus		Mandatory	See Table 3 - OperationalStatus for ComputerSystem.
IdentifyingDescriptions		Mandatory	'DomainID' is placed into the entry that corresponds to the DomainID in OtherIdentifyingInfo and 'SNIA:DetailedType' is placed in the entry corresponding to 'Virtual Switch' in OtherIdentifyingInfo.
Dedicated		Mandatory	See the Dedicated definition in 13.8.3 CIM_ComputerSystem (Switch).
EnabledState		Conditional	Conditional requirement: Support for the Fabric profile. See the EnabledState definition in 5.8.10 CIM_ComputerSystem (Switch).
RequestedState		Mandatory	See the RequestedState definition in 13.8.3 CIM_ComputerSystem (Switch).
EnabledDefault		Optional	See the EnabledDefault definition in 13.8.3 CIM_ComputerSystem (Switch).
RequestStateChange()		Conditional	Conditional requirement: Support for a non-null value in FCSwitchCapabilities.RequestedStatesSupported.

13.8.3 CIM_ComputerSystem (Switch)

Represents the Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Shall be associated to RegisteredProfile using ElementConformsToProfile association. The RegisteredProfile instance shall have RegisteredName set to 'Switch', RegisteredOrganization set to 'SNIA', and RegisteredVersion set to '1.5.0'.

Table 135 describes class CIM_ComputerSystem (Switch).

Table 135 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Switch)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name	D	Mandatory	Switch Name (WWN).
ElementName		Conditional	Conditional requirement: Support for the Fabric profile. See the ElementName definition in 5.8.10 CIM_ComputerSystem (Switch). User-friendly name. Can be set if FCSwitchCapabilities.ElementNameEditSupported for the switch is True.
NameFormat		Mandatory	Shall be 'WWN'.
OtherIdentifyingInfo	C	Conditional	Conditional requirement: Support for the Fabric profile. See the OtherIdentifyingInfo definition in 5.8.10 CIM_ComputerSystem (Switch). DomainID stored in decimal format, with a value between 0 and 255. In addition, if the Fabric profile is implemented, then another value corresponding to 'SNIA:DetailedType' shall contain 'Front Domain', 'Translate Domain', 'Virtual Switch', 'Backbone' or 'None'.
OperationalStatus		Mandatory	See Table 3 - OperationalStatus for ComputerSystem.
IdentifyingDescriptions		Conditional	Conditional requirement: Support for the Fabric profile. See the IdentifyingDescriptions definition in 5.8.10 CIM_ComputerSystem (Switch). 'DomainID' is placed into corresponding index of OtherIdentifyingInfo. In addition, if the Fabric profile is implemented, then one of the other indices shall contain the value 'SNIA:DetailedType' and in the corresponding index for OtherIdentifyingInfo a value of 'Front Domain' for an Inter-Fabric Routing (IFR) Front Domain, 'Translate Domain' for an Inter-Fabric Routing (IFR) Translate Domain, 'Virtual Switch' for a Partitioned Switch, 'Backbone' for a Backbone Switch, or 'None' when the other values don't apply.
Dedicated		Mandatory	'Switch'.
EnabledState		Conditional	Conditional requirement: Support for the Fabric profile. See the EnabledState definition in 5.8.10 CIM_ComputerSystem (Switch).
RequestedState		Mandatory	The Switch state requested via RequestStateChange(). Shall be of the range specified in FCSwitchCapabilities.RequestedStatesSupported if a state change has been requested. Otherwise shall be 'Not Applicable'.
EnabledDefault		Optional	Default startup for the Switch.
RequestStateChange()		Conditional	Conditional requirement: Support for a non-null value in FCSwitchCapabilities.RequestedStatesSupported.

13.8.4 CIM_ComputerSystemPackage

Associates PhysicalPackage to the ComputerSystem (Switch).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 136 describes class CIM_ComputerSystemPackage.

Table 136 - SMI Referenced Properties/Methods for CIM_ComputerSystemPackage

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The reference to the PhysicalPackage.
Dependent		Mandatory	The reference to the Switch ComputerSystem.

13.8.5 CIM_ElementCapabilities (FCPort to FCPortCapabilities)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 137 describes class CIM_ElementCapabilities (FCPort to FCPortCapabilities).

Table 137 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (FCPort to FCPortCapabilities)

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	The reference to the switch FCPort.
Capabilities		Mandatory	The reference to the FCPortCapabilities.

13.8.6 CIM_ElementCapabilities (System to FCSwitchCapabilities)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 138 describes class CIM_ElementCapabilities (System to FCSwitchCapabilities).

Table 138 - SMI Referenced Properties/Methods for CIM_ElementCapabilities (System to FCSwitchCapabilities)

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	The reference to the Switch ComputerSystem.
Capabilities		Mandatory	The reference to the FCSwitchCapabilities.

13.8.7 CIM_ElementSettingData (FCPortSettings to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 139 describes class CIM_ElementSettingData (FCPortSettings to FCPort).

Table 139 - SMI Referenced Properties/Methods for CIM_ElementSettingData (FCPortSettings to FCPort)

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	The reference to the switch FCPort.
SettingData		Mandatory	The reference to the FCPortSettings.

13.8.8 CIM_ElementSettingData (FCSwitchSettings to ComputerSystem)

Associates FCSwitchSettings to ComputerSystem (Switch).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Support for the Switch Configuration Data profile.

Table 140 describes class CIM_ElementSettingData (FCSwitchSettings to ComputerSystem).

Table 140 - SMI Referenced Properties/Methods for CIM_ElementSettingData (FCSwitchSettings to ComputerSystem)

Properties	Flags	Requirement	Description and Notes
SettingData		Mandatory	The reference to the FCSwitchSettings.
ManagedElement		Mandatory	The reference to the switch ComputerSystem.

13.8.9 CIM_ElementStatisticalData (FCPortRateStatistics to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 141 describes class CIM_ElementStatisticalData (FCPortRateStatistics to FCPort).

Table 141 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPortRateStatistics to FCPort)

Properties	Flags	Requirement	Description and Notes
Stats		Mandatory	The reference to the FCPortRateStatistics.
ManagedElement		Mandatory	The reference to the switch FCPort.

13.8.10 CIM_ElementStatisticalData (FCPortStatistics to FCPort)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 142 describes class CIM_ElementStatisticalData (FCPortStatistics to FCPort).

Table 142 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPortStatistics to FCPort)

Properties	Flags	Requirement	Description and Notes
Stats		Mandatory	The reference to the FCPortStatistics.
ManagedElement		Mandatory	The reference to the switch FCPort.

13.8.11 CIM_FCPort (Partitioned Switch FCPort)

The Fibre Channel Switch Port for a Virtual Switch.

Created By: Static

Modified By: Extrinsic: RequestStateChange

Deleted By: Static

Requirement: Support for the Switch Partitioning profile.

Table 143 describes class CIM_FCPort (Partitioned Switch FCPort).

Table 143 - SMI Referenced Properties/Methods for CIM_FCPort (Partitioned Switch FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in 13.8.12 CIM_FCPort (Switch FCPort).
SystemName		Mandatory	See the SystemName definition in 13.8.12 CIM_FCPort (Switch FCPort).
CreationClassName		Mandatory	See the CreationClassName definition in 13.8.12 CIM_FCPort (Switch FCPort).
DeviceID		Mandatory	See the DeviceID definition in 13.8.12 CIM_FCPort (Switch FCPort).
ElementName		Mandatory	See the ElementName definition in 13.8.12 CIM_FCPort (Switch FCPort).
OperationalStatus		Mandatory	See Table 2 - Port OperationalStatus.
EnabledState		Conditional	Conditional requirement: Support for the Fabric profile. See clause 5.
DetailedPortState		Mandatory	Experimental. See Table 130 - DetailedPortState for FCPort.
PortAvailability		Optional	Experimental. See Table 131 - PortAvailability for FCPort.
RequestedState		Mandatory	See the RequestedState definition in 13.8.12 CIM_FCPort (Switch FCPort).
EnabledDefault		Mandatory	See the EnabledDefault definition in 13.8.12 CIM_FCPort (Switch FCPort).
Speed		Mandatory	See the Speed definition in 13.8.12 CIM_FCPort (Switch FCPort).
MaxSpeed		Mandatory	See the MaxSpeed definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortType		Mandatory	See the PortType definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortNumber		Mandatory	See the PortNumber definition in 13.8.12 CIM_FCPort (Switch FCPort).

Table 143 - SMI Referenced Properties/Methods for CIM_FCPort (Partitioned Switch FCPort)

Properties	Flags	Requirement	Description and Notes
PermanentAddress		Mandatory	See the PermanentAddress definition in 13.8.11 CIM_FCPort (Switch FCPort).
LinkTechnology		Mandatory	See the LinkTechnology definition in 13.8.12 CIM_FCPort (Switch FCPort).
PortDiscriminator		Conditional	Experimental. Conditional requirement: Support for the Switch Partitioning profile. This array property identifies the context in which this FCPort is instantiated. For Partitioned Switch FCPorts, the values may be '3' (VF), '6' (IFR (Virtual)) or '7' (NPIV).
RequestStateChange()		Conditional	Conditional requirement: Support for a non-null value in FCPortCapabilities.RequestedStatesSupported. See the RequestedStateChange definition in 13.8.12 CIM_FCPort (Switch FCPort).

13.8.12 CIM_FCPort (Switch FCPort)

The Fibre Channel Switch Port.

Created By: Static

Modified By: Extrinsic: RequestStateChange

Deleted By: Static

Requirement: Mandatory.

Table 144 describes class CIM_FCPort (Switch FCPort).

Table 144 - SMI Referenced Properties/Methods for CIM_FCPort (Switch FCPort)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	The Class Name.
DeviceID		Mandatory	Opaque.
ElementName		Mandatory	User-friendly name. Can be set if FCPortCapabilities.ElementNameEditSupported is True.
OperationalStatus		Mandatory	See Table 2 - Port OperationalStatus.
EnabledState		Conditional	Conditional requirement: Support for the Fabric profile. See clause 5.
DetailedPortState		Mandatory	Experimental. See Table 130 - DetailedPortState for FCPort.
PortAvailability		Optional	Experimental. See Table 131 - PortAvailability for FCPort.
RequestedState		Mandatory	The port state requested via RequestStateChange(). Shall be of the range specified in FCPortCapabilities.RequestedStatesSupported if a state change has been requested. Otherwise shall be 'Not Applicable'.
EnabledDefault		Mandatory	Default startup for the port. Used in conjunction with RequestedState can allow for persistent disabling of a port.

Table 144 - SMI Referenced Properties/Methods for CIM_FCPort (Switch FCPort)

Properties	Flags	Requirement	Description and Notes
Speed		Mandatory	Speed of zero represents a link not established. 1 GB/s is 1 062 500 000 bit/s 2 GB/s is 2 125 000 000 bit/s 4 GB/s is 4 250 000 000 bit/s 10 GB/s single channel variants are 10 518 750 000 bit/s 10 GB/s four channel variants are 12 750 000 000 bit/s This is the raw bit rate.
MaxSpeed		Mandatory	The max speed of the Port in bits per second using the same algorithm as Speed.
PortType		Mandatory	FC-GS-6 Port.Type. The specific mode currently enabled for the Port. The values: 'N' = Node Port 'NL' = Node Port supporting FC arbitrated loop 'E' = Expansion Port connecting fabric elements (for example, FC switches) 'F' = Fabric (element) Port 'EL' = Fabric (element) Port supporting FC arbitrated loop 'B' = Bridge Port. PortTypes are defined in the FC-GS-6 standards. Can be set using FCPortSettings.RequestedType.
PortNumber		Mandatory	NetworkPorts are often numbered relative to either a logical module or a network element.
PermanentAddress		Mandatory	Fibre Channel Port WWN.
LinkTechnology		Mandatory	'FC'.
PortDiscriminator		Optional	Experimental. This array property identifies the context in which this FCPort is instantiated. For this standard, the values may be '0' (Unknown), '2' (Not applicable), '3' (VF), '4' (FCIP), '5' (IFR), '6' (IFR (Virtual)), '7' (NPIV), '8' (Internal) or '9' (Chassis).
RequestStateChange()		Conditional	Conditional requirement: Support for a non-null value in FCPortCapabilities.RequestedStatesSupported. Method to change the port state. FCPortCapabilities.RequestedStatesSupported indicates what states can be set.

13.8.13 CIM_FCPortCapabilities

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 145 describes class CIM_FCPortCapabilities.

Table 145 - SMI Referenced Properties/Methods for CIM_FCPortCapabilities

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	Shall be 'FC Port Capabilities'.
ElementNameEditSupported		Mandatory	Indicates whether FCPort.ElementName is settable.
MaxElementNameLen		Mandatory	Indicates the maximum string length of FCPort.ElementName.
RequestedStatesSupported	N	Mandatory	Indicates the supported states for calling FCPort.RequestStateChange(). If this property is NULL it means RequestedStateChange is not supported. Otherwise the values may be '2' (Enabled), '3' (Disabled), '4' (Shutdown), '6' (Offline), '7' (Test), '8' (Defer), '9' (Quiesce), '10' (Reboot) and/or '11' (Reset). If either '2' (Enabled) or '3' (Disabled) is supported, then both '2' (Enabled) and '3' (Disabled) shall be supported.
RequestedSpeedsSupported		Mandatory	Indicates the supported speeds that can be set in FCPortSettings.RequestedSpeed.
AutoSenseSpeedConfigurable		Mandatory	Indicates whether FCPortSettings.AutoSenseSpeed can be set to auto-negotiate speed.
RequestedTypesSupported		Mandatory	Indicates the list of supported port types that can be set in FCPortSettings.RequestedType.
NPIVSupported		Optional	Experimental.
MaxNPIVLogins		Optional	Experimental.

13.8.14 CIM_FCPortRateStatistics

Fibre Channel Switch Port Rate Statistics represent the rate per second over the SampleInterval. An instance of this class can represent the statistics for the current statistics, archived and consolidated statistics, or both.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 146 describes class CIM_FCPortRateStatistics.

Table 146 - SMI Referenced Properties/Methods for CIM_FCPortRateStatistics

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
StatisticTime		Mandatory	The time the statistic was collected.
SampleInterval		Mandatory	The interval at which the rates are calculated.
TxFramerate		Optional	
RxFramerate		Optional	
MaxTxFramerate		Optional	
MaxRxFramerate		Optional	
TxRate		Mandatory	

Table 146 - SMI Referenced Properties/Methods for CIM_FCPortRateStatistics

Properties	Flags	Requirement	Description and Notes
RxRate		Mandatory	
PeakTxRate		Optional	
PeakRxRate		Optional	

13.8.15 CIM_FCPortSettings

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 147 describes class CIM_FCPortSettings.

Table 147 - SMI Referenced Properties/Methods for CIM_FCPortSettings

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	Shall be 'FC Port Settings'.
RequestedSpeed	M	Mandatory	The requested value to which FCPort.Speed should be set.
AutoSenseSpeed	M	Mandatory	The request for the FCPort to auto sense the speed (FCPort.Speed).
RequestedType	M	Mandatory	The requested setting for the FCPort.PortType.

13.8.16 CIM_FCPortStatistics

Snapshot of performance and error counters for the Fibre Channel Switch.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 148 describes class CIM_FCPortStatistics.

Table 148 - SMI Referenced Properties/Methods for CIM_FCPortStatistics

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
StatisticTime		Optional	The time the statistics were collected. If historical data is instantiated (present), this property shall be set with the time representing the time the statistic was collected.
ElementName		Optional	
BytesTransmitted		Mandatory	
BytesReceived		Mandatory	
PacketsTransmitted		Mandatory	
PacketsReceived		Mandatory	
CRCErrors		Mandatory	

Table 148 - SMI Referenced Properties/Methods for CIM_FCPortStatistics

Properties	Flags	Requirement	Description and Notes
LinkFailures		Mandatory	
PrimitiveSeqProtocolErrCount		Mandatory	
LIPCount		Optional	
NOSCount		Optional	
ErrorFrames		Optional	
DumpedFrames		Optional	
LossOfSignalCounter		Optional	
LossOfSyncCounter		Optional	
InvalidTransmissionWords		Optional	
FramesTooShort		Optional	
FramesTooLong		Optional	
AddressErrors		Optional	
BufferCreditNotProvided		Optional	
BufferCreditNotReceived		Optional	
DelimiterErrors		Optional	
EncodingDisparityErrors		Optional	
LinkResetsReceived		Optional	
LinkResetsTransmitted		Optional	
MulticastFramesReceived		Optional	
MulticastFramesTransmitted		Optional	
FBSYFrames		Optional	
PBSYFrames		Optional	
FRJTFrames		Optional	
PRJTFrames		Optional	
RXClass1Frames		Optional	
TXClass1Frames		Optional	
Class1FBSY		Optional	
Class1PBSY		Optional	
Class1FRJT		Optional	
Class1PRJT		Optional	
RXClass2Frames		Optional	
TXClass2Frames		Optional	
Class2FBSY		Optional	
Class2PBSY		Optional	
Class2FRJT		Optional	
Class2PRJT		Optional	

Table 148 - SMI Referenced Properties/Methods for CIM_FCPortStatistics

Properties	Flags	Requirement	Description and Notes
RXClass3Frames		Optional	
TXClass3Frames		Optional	
Class3FramesDiscarded		Optional	
RXBroadcastFrames		Optional	
TXBroadcastFrames		Optional	
FramePacingCount		Optional	Experimental. The number of 2.5 us units that a frame transmission is blocked due to buffer-to-buffer credits.

13.8.17 CIM_FCSwitchCapabilities

Fibre Channel Switch Capabilities.

EXPERIMENTAL |

When the attached HBA supports FC-SB-4, and the switch is in a cascaded configuration, DomainIDConfigureable and DomainIDLockedSupported shall be TRUE.

| EXPERIMENTAL.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 149 describes class CIM_FCSwitchCapabilities.

Table 149 - SMI Referenced Properties/Methods for CIM_FCSwitchCapabilities

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	Shall be 'FC Switch Capabilities'.
ElementNameEditSupported		Mandatory	Capability indicating whether ComputerSystem.ElementName for the switch can be set.
MaxElementNameLen		Mandatory	Capability specifying the maximum name of ComputerSystem.ElementName for the switch.
RequestedStatesSupported	N	Mandatory	The states the switch can support via ComputerSystem.RequestedState. If this property is NULL it means RequestedStateChange is not supported. Otherwise the values may be '2' (Enabled), '3' (Disabled), '4' (Shutdown), '6' (Offline), '7' (Test), '8' (Defer), '9' (Quiesce), '10' (Reboot) and/or '11' (Reset).
DomainIDConfigureable		Mandatory	Indicates whether the DomainID setting can be modified.
MinDomainID		Conditional	Conditional requirement: Support for DomainID configuration. Shall be set if DomainIDConfigureable is true.
MaxDomainID		Conditional	Conditional requirement: Support for DomainID configuration. Shall be set if DomainIDConfigureable is true.

Table 149 - SMI Referenced Properties/Methods for CIM_FCSwitchCapabilities

Properties	Flags	Requirement	Description and Notes
DomainIDLockedSupported		Mandatory	
PrincipalPrioritiesSupported		Mandatory	
NPIVSupported		Optional	Experimental.

13.8.18 CIM_FCSwitchSettings

Fibre Channel Switch Settings.

EXPERIMENTAL |

When the attached HBA supports FC-SB, and the switch is in a cascaded configuration, PreferredDomainID shall be set to a unique value, and DomainIDLocked shall be true.

| EXPERIMENTAL.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 150 describes class CIM_FCSwitchSettings.

Table 150 - SMI Referenced Properties/Methods for CIM_FCSwitchSettings

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	Shall be 'FC Switch Settings'.
PreferredDomainID	M	Conditional	Conditional requirement: Support for DomainID configuration. Required if FCSwitchCapabilities.DomainIDConfigureable is TRUE.
DomainIDLocked	M	Conditional	Conditional requirement: Support for DomainID locking. Required if FCSwitchCapabilities.DomainIDLockSupported is TRUE.
PrincipalPriority	M	Conditional	Conditional requirement: Support for Principal Priorities. Required if FCSwitchCapabilities.PrincipalPrioritiesSupported is not set to 'Not Applicable'.

13.8.19 CIM_FilterCollection (Switch Predefined FilterCollection)

Experimental. This is a collection of predefined IndicationFilters to which a client may subscribe. A switch implementation shall indicate support for predefined FilterCollections by the SNIA_IndicationConfigurationCapabilities.FeaturesSupported = '5' (Predefined Filter Collections).

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 151 describes class CIM_FilterCollection (Switch Predefined FilterCollection).

**Table 151 - SMI Referenced Properties/Methods for CIM_FilterCollection
(Switch Predefined FilterCollection)**

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Shall specify the unique identifier for an instance of this class within the Implementation namespace.
CollectionName		Mandatory	The value of CollectionName shall be 'SNIA:Switch'.

13.8.20 CIM_HostedCollection (Redundancy Set)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 152 describes class CIM_HostedCollection (Redundancy Set).

**Table 152 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Redundancy Set)**

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The switch ComputerSystem.
Dependent		Mandatory	The redundancy set of FCPorts.

13.8.21 CIM_HostedCollection (Statistics Collection)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 153 describes class CIM_HostedCollection (Statistics Collection).

**Table 153 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Statistics Collection)**

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The switch ComputerSystem.
Dependent		Mandatory	The Statistics Collection.

13.8.22 CIM_HostedCollection (Switch to predefined FilterCollection)

Experimental.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 154 describes class CIM_HostedCollection (Switch to predefined FilterCollection).

**Table 154 - SMI Referenced Properties/Methods for CIM_HostedCollection
(Switch to predefined FilterCollection)**

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the predefined FilterCollection for the Switch.
Antecedent		Mandatory	Reference to the Switch ComputerSystem.

13.8.23 CIM_IndicationFilter (Switch ComputerSystem Creation)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new switch computer system instance. This would represent the addition of a switch computer system. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures=3 (Predefined Filters).

Table 155 describes class CIM_IndicationFilter (Switch ComputerSystem Creation).

**Table 155 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Switch ComputerSystem Creation)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Switch:ComputerSystemCreation'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

13.8.24 CIM_IndicationFilter (Switch ComputerSystem Deletion)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a switch computer system instance. This would represent the removal of a switch computer system. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 156 describes class CIM_IndicationFilter (Switch ComputerSystem Deletion).

Table 156 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Switch ComputerSystem Deletion)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Switch:ComputerSystemDeletion'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

13.8.25 CIM_IndicationFilter (Switch ComputerSystem OperationalStatus)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a switch ComputerSystem instance. This would represent the status change to a switch computer system. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 157 describes class CIM_IndicationFilter (Switch ComputerSystem OperationalStatus).

Table 157 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Switch ComputerSystem OperationalStatus)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Switch:ComputerSystemOperationalStatus'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

13.8.26 CIM_IndicationFilter (Switch FCPort OperationalStatus)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a switch FCPort instance. This would represent the status change to a switch FCPort. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 158 describes class CIM_IndicationFilter (Switch FCPort OperationalStatus).

**Table 158 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Switch FCPort OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in section <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Switch:FCPortOperationalStatus'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

13.8.27 CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus)

Deprecated. This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a switch computer system instance. This would represent the OperationalStatus changes of a switch computer system. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 159 describes class CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus).

Table 159 - SMI Referenced Properties/Methods for CIM_IndicationFilter (WQL Switch ComputerSystem OperationalStatus)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Switch:ComputerSystemOperationalStatusWQL'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus.
QueryLanguage		Mandatory	This shall be 'WQL' or 'SMI-S V1.0'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

13.8.28 CIM_IndicationFilter (WQL Switch FCPort OperationalStatus)

Deprecated. This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a switch FCPort instance. This would represent the OperationalStatus changes of a switch FCPort. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters).

Table 160 describes class CIM_IndicationFilter (WQL Switch FCPort OperationalStatus).

**Table 160 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(WQL Switch FCPort OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Switch:FCPortOperationalStatusWQL'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus.
QueryLanguage		Mandatory	This shall be 'WQL' or 'SMI-S V1.0'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

13.8.29 CIM_MemberOfCollection (FCPort to RedundancySet)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 161 describes class CIM_MemberOfCollection (FCPort to RedundancySet).

**Table 161 - SMI Referenced Properties/Methods for CIM_MemberOfCollection
(FCPort to RedundancySet)**

Properties	Flags	Requirement	Description and Notes
Member		Mandatory	The reference to the switch FCPort.
Collection		Mandatory	The reference to the RedundancySet.

13.8.30 CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 162 describes class CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection).

Table 162 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (NetworkPortStatistics to StatisticalCollection)

Properties	Flags	Requirement	Description and Notes
Member		Mandatory	The reference to the FCPortStatistics.
Collection		Mandatory	The reference to the StatisticsCollection.

13.8.31 CIM_MemberOfCollection (Predefined Filter Collection to Switch Filters)

Experimental. This associates the switch predefined FilterCollection to the predefined Filters supported by the switch.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 163 describes class CIM_MemberOfCollection (Predefined Filter Collection to Switch Filters).

Table 163 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Switch Filters)

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the switch predefined FilterCollection.
Member		Mandatory	Reference to the predefined IndicationFilters of the switch.

13.8.32 CIM_ProtocolEndpoint

The endpoint of a link (ActiveConnection). ProtocolEndpoint shall be implemented when BroadcastReset() is supported (Force LIP). It is expected that the Fabric Profile is also implemented which defines the necessary information for determining who will receive the Force LIP on the loop.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 164 describes class CIM_ProtocolEndpoint.

Table 164 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
Name	CD	Mandatory	The Fibre Channel Port WWN.
NameFormat		Mandatory	'WWN'.

Table 164 - SMI Referenced Properties/Methods for CIM_ProtocolEndpoint

Properties	Flags	Requirement	Description and Notes
ProtocolType		Mandatory	Shall be 56 (Fibre Channel).
BroadcastResetSupported		Mandatory	
BroadcastReset()		Optional	Sends a Force LIP to all attached Ports. Required if BroadcastResetSupported is TRUE.

13.8.33 CIM_RedundancySet

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 165 describes class CIM_RedundancySet.

Table 165 - SMI Referenced Properties/Methods for CIM_RedundancySet

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
TypeOfSet		Mandatory	

13.8.34 CIM_StatisticsCollection

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 166 describes class CIM_StatisticsCollection.

Table 166 - SMI Referenced Properties/Methods for CIM_StatisticsCollection

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	
SampleInterval		Mandatory	
TimeLastSampled		Mandatory	

13.8.35 CIM_SystemDevice

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 167 describes class CIM_SystemDevice.

Table 167 - SMI Referenced Properties/Methods for CIM_SystemDevice

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	The reference to the switch FCPort.
GroupComponent		Mandatory	The reference to the switch System.

STABLE

IECNORM.COM : Click to view the full PDF of ISO/IEC 24775-6:2014

EXPERIMENTAL

14 Switch Configuration Data Subprofile

14.1 Description

This subprofile describes the ability to retrieve a configuration from a switch and latter apply that configuration back on the switch (similar to an image backup and restoration of a computer system).

The profile only has three classes providing all the functionality. When a client needs to obtain a snapshot of the switch configuration, he/she enumerates ConfigurationData which will return the current configuration with the timestamp set appropriately.

When the client wants to apply a configuration, the client creates an instance of ConfigurationData and calls the method ApplyConfiguration() on the instance containing the property ConfigurationInformation which is to be applied to the switch. Figure 23 shows a Switch Configuration Data instance.

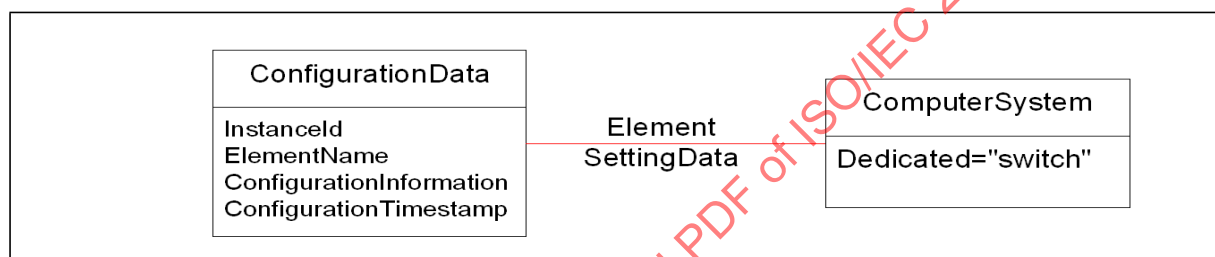


Figure 23 - Switch Configuration Data Instance

14.2 Durable Names and Correlatable IDs of the Profile

None in this International Standard.

14.3 Instrumentation Requirements

None in this International Standard.

14.4 Health and Fault Management

None.

14.5 Cascading Considerations

None.

14.6 Methods of this Profile – ApplyConfiguration

This method applies the configuration data to the switch. The data in the instance's ConfigurationInformation property is used as the configuration to apply. Note that it is not necessary for the element to be associated with the ConfigurationData instance at the time that this method is called.

```

uint32 ApplyConfiguration (
    boolean ValidateOnly,
    uint16 TypeOfConfiguration
    CIM_ManagedElement REF ManagedElement);
  
```

14.7 Client Considerations and Recipes

14.7.1 Get Switch Configuration

```

// DESCRIPTION
//
  
```

```
// This recipe describes how to retrieve Switch configuration data.
//
// PRE-EXISTING CONDITIONS AND ASSUMPTIONS
// 1. A reference to the Switch whose configuration data to retrieve is
//    known
// and defined in the variable $Switch->.

// MAIN
// Step 1. Retrieve the configuration of the Switch.
$ConfigData[] = Associators($Switch->,
    "CIM_ElementSettingData",
    "CIM_ConfigurationData",
    "ManagedElement",
    "SettingData",
    false,
    false,
    {"ConfigurationInformation", "ConfigurationTimestamp"})
if ($ConfigData[] == null || $ConfigData[].length != 1) {
    <ERROR! The required Switch configuration data is not available>
}
$SwitchConfig = $ConfigData[0]
```

14.7.2 Set Switch Configuration

```
// DESCRIPTION
//
// Set Switch Configuration
//
// PREEXISTING CONDITIONS AND ASSUMPTIONS
//
// None

Placeholder File
```

14.8 Registered Name and Version

Switch Configuration Data version 1.1.0 (Component Profile).

14.9 CIM Elements

14.9.1 General

Table 168 describes the CIM elements for Switch Configuration Data.

Table 168 - CIM Elements for Switch Configuration Data

Element Name	Requirement	Description
14.9.2 CIM_ComputerSystem	Mandatory	Represents the Switch.
14.9.3 CIM_ConfigurationData	Mandatory	Switch Configuration Data.
14.9.4 CIM_ElementSettingData	Mandatory	Associates ConfigurationData to the switch.

14.9.2 CIM_ComputerSystem

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 169 describes class CIM_ComputerSystem.

Table 169 - SMI Referenced Properties/Methods for CIM_ComputerSystem

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	The class name.
Name		Mandatory	Switch Name (WWN).
NameFormat		Mandatory	"WWN".
Dedicated		Mandatory	"Switch".

14.9.3 CIM_ConfigurationData

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 170 describes class CIM_ConfigurationData.

Table 170 - SMI Referenced Properties/Methods for CIM_ConfigurationData

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	User friendly for configuration file.
ConfigurationInformation		Mandatory	The configuration data of the switch.
ConfigurationTimestamp		Mandatory	Time the configuration data was obtained.
ApplyConfiguration()		Mandatory	Method that processes the configuration in the same instance and applies it to the switch.

14.9.4 CIM_ElementSettingData

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 171 describes class CIM_ElementSettingData.

Table 171 - SMI Referenced Properties/Methods for CIM_ElementSettingData

Properties	Flags	Requirement	Description and Notes
SettingData		Mandatory	
ManagedElement		Mandatory	

EXPERIMENTAL

IECNORM.COM : Click to view the full PDF of ISO/IEC 24775-6:2014

STABLE

15 Blades Subprofile

15.1 Description

15.1.1 General

This subprofile describes how blades in a director class switch can be discovered and managed.

15.1.2 Instance Diagram

Figure 24 shows a Switch Blade instance.

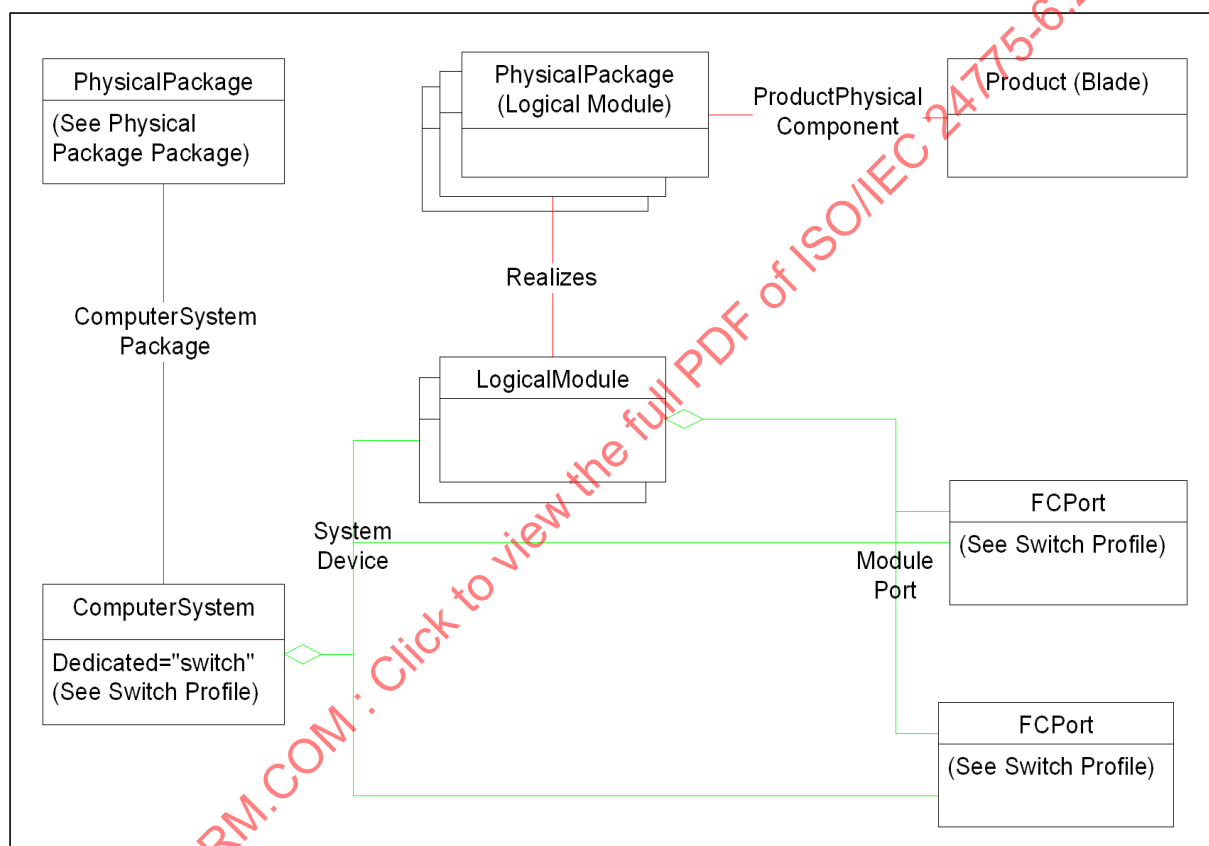


Figure 24 - Switch Blade Instance

EXPERIMENTAL

15.1.3 Blades Predefined Indications

If the optional Experimental Indication profile is supported by an implementation, there shall be an implementation of the SNIA_IndicationConfigurationService and its associated SNIA_IndicationConfigurationCapabilities associated to the Switch ComputerSystem associated with the Blades profile. If the implementation supports predefined IndicationFilters or predefined IndicationFilterCollections this shall be indicated in the SupportedFeatures property of the SNIA_IndicationConfigurationCapabilities. If a value "3" is present, it means the implementation supports predefined IndicationFilters. If a value of "5" is present, it means the implementation supports predefined IndicationFilterCollections.

Figure 25 illustrates classes that shall be populated by the Blades profile if both "3" and "5" are present in the SupportedFeatures property.

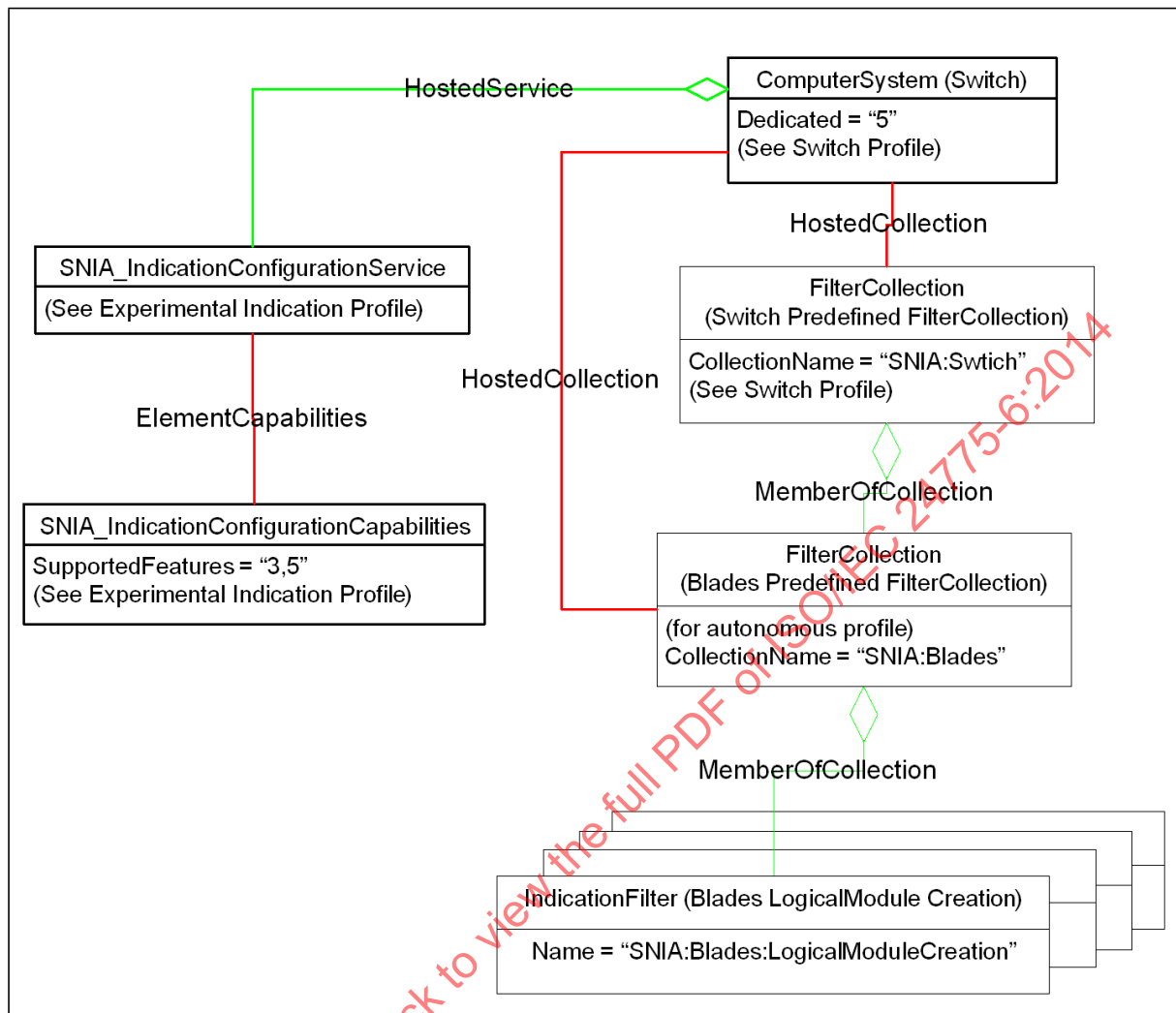


Figure 25 - Predefined Blades IndicationFilters and Collection

The SNIA_IndicationConfigurationService is hosted on the switch ComputerSystem for the Switch Profile associated to the Blades component profile. The FilterCollection for blades is also hosted on the same switch ComputerSystem. The blades FilterCollection is a member (MemberOfCollection) in the switch FilterCollection. The blades FilterCollection has members which are all the predefined IndicationFilters supported by the implementation. This shall include all Mandatory IndicationFilters of the Blades Profile. But it should also include any IndicationFilter that has been predefined by the implementation. This may include conditional, optional or vendor specific IndicationFilters supported by the implementation.

The blades FilterCollection shall have the CollectionName "SNIA:Blades". Each of the predefined filters shall have the Name property as defined for the IndicationFilter. In Figure 25 the name of the IndicationFilter (Blades LogicalModule Creation) is "SNIA:Blades:LogicalModuleCreation". For vendor specific IndicationFilters (not defined in this International Standard), the Name of the filter would be of the form ORG_ID":Blades:"UNIQUE_ID, where ORGID is the designation of the vendor that is providing the implementation.

EXPERIMENTAL

15.2 Health and Fault Management

None.

15.3 Cascading Considerations

None.

15.4 Dependencies on Profiles, Subprofiles, and Packages

Related Profiles for Blades: Not defined in this International Standard.

15.5 Methods of this Profile

None.

15.6 Client Considerations and Recipes

None.

15.7 Registered Name and Version

Blades version 1.5.0 (Component Profile).

15.8 CIM Elements**15.8.1 General**

Table 172 describes the CIM elements for Blades.

Table 172 - CIM Elements for Blades

Element Name	Requirement	Description
15.8.2 CIM_FilterCollection (Blades Predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This is a collection of predefined IndicationFilters to which a client may subscribe.
15.8.3 CIM_HostedCollection (Switch to Blades predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).
15.8.4 CIM_IndicationFilter (Blades LogicalModule Creation)	Optional	Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new blade LogicalModule instance.
15.8.5 CIM_IndicationFilter (Blades LogicalModule Deletion)	Optional	Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a blade LogicalModule instance.
15.8.6 CIM_IndicationFilter (Blades LogicalModule OperationalStatus)	Optional	Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the modification of OperationalStatus of a blade LogicalModule instance.
15.8.7 CIM_IndicationFilter (WQL Blades LogicalModule OperationalStatus)	Optional	Deprecated. This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a blade LogicalModule instance.
15.8.8 CIM_LogicalModule	Mandatory	The Blade.
15.8.9 CIM_MemberOfCollection (Blades FilterCollection to Switch FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This associates the Blades predefined FilterCollection to the Switch predefined FilterCollection.

Table 172 - CIM Elements for Blades

Element Name	Requirement	Description
15.8.10 CIM_MemberOfCollection (Predefined Filter Collection to Blades Filters)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This associates the Blades predefined FilterCollection to the predefined Filters supported by the Blade.
15.8.11 CIM_ModulePort	Mandatory	Associates the LogicalModule to the FCPort.
15.8.12 CIM_PhysicalPackage (Logical Module)	Mandatory	The physical package within which the LogicalModule is contained.
15.8.13 CIM_Product (Blade)	Optional	The product information for the Blade.
15.8.14 CIM_ProductPhysicalComponent	Optional	Associates the Product to the PhysicalPackage.
15.8.15 CIM_Realizes (Logical Module Package)	Optional	Associates the LogicalModule to its PhysicalPackage.
15.8.16 CIM_SystemDevice (Logical Module)	Mandatory	Associates the LogicalModule to the ComputerSystem representing the Switch.
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_LogicalModule	Optional	Creation of an Creation LogicalModule instance. See 15.8.4 CIM_IndicationFilter (Blades LogicalModule Creation).
SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_LogicalModule	Optional	Deletion of an LogicalModule instance. See 15.8.5 CIM_IndicationFilter (Blades LogicalModule Deletion).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalModule AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Optional	Deprecated. Deprecated WQL - Change in status of LogicalModule. See 15.8.7 CIM_IndicationFilter (WQL Blades LogicalModule OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalModule AND SourceInstance.CIM_LogicalModule::OperationalStatus <> PreviousInstance.CIM_LogicalModule::OperationalStatus	Optional	CQL - Change in status of LogicalModule. See 15.8.6 CIM_IndicationFilter (Blades LogicalModule OperationalStatus).

15.8.2 CIM_FilterCollection (Blades Predefined FilterCollection)

Experimental. This is a collection of predefined IndicationFilters to which a client may subscribe. A blades implementation shall indicate support for predefined FilterCollections by the SNIA_IndicationConfigurationCapabilities.FeaturesSupported = '5' (Predefined Filter Collections).

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 173 describes class CIM_FilterCollection (Blades Predefined FilterCollection).

Table 173 - SMI Referenced Properties/Methods for CIM_FilterCollection (Blades Predefined FilterCollection)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Shall specify the unique identifier for an instance of this class within the Implementation namespace.
CollectionName		Mandatory	The value of CollectionName shall be 'SNIA:Blades'.

15.8.3 CIM_HostedCollection (Switch to Blades predefined FilterCollection)

Experimental.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 174 describes class CIM_HostedCollection (Switch to Blades predefined FilterCollection).

Table 174 - SMI Referenced Properties/Methods for CIM_HostedCollection (Switch to Blades predefined FilterCollection)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	Reference to the predefined FilterCollection for the Blade.
Antecedent		Mandatory	Reference to the Switch ComputerSystem.

15.8.4 CIM_IndicationFilter (Blades LogicalModule Creation)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new blade LogicalModule instance. This would represent the addition of a blade LogicalModule. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 175 describes class CIM_IndicationFilter (Blades LogicalModule Creation).

Table 175 - SMI Referenced Properties/Methods for CIM_IndicationFilter (Blades LogicalModule Creation)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Blades:LogicalModuleCreation'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 175 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Blades LogicalModule Creation)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_LogicalModule.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

15.8.5 CIM_IndicationFilter (Blades LogicalModule Deletion)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the removal of a blade LogicalModule instance. This would represent the removal of a blade LogicalModule. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 176 describes class CIM_IndicationFilter (Blades LogicalModule Deletion).

**Table 176 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Blades LogicalModule Deletion)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Blades:LogicalModuleDeletion'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 176 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Blades LogicalModule Deletion)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_LogicalModule.
QueryLanguage		Mandatory	This should be 'DMTF:CQL' for CQL queries, but may be 'WQL' or 'SMI-S V1.0'. WQL and SMI-S V1.0 are deprecated in favor of 'DMTF:CQL'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

15.8.6 CIM_IndicationFilter (Blades LogicalModule OperationalStatus)

Experimental. This is the 'pre-defined' CIM_IndicationFilter instance for the change of OperationalStatus of a blade LogicalModule instance. This would represent the status change to a blade LogicalModule. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 177 describes class CIM_IndicationFilter (Blades LogicalModule OperationalStatus).

**Table 177 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Blades LogicalModule OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Blades:LogicalModuleOperationalStatus'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 177 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(Blades LogicalModule OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalModule AND SourceInstance.CIM_LogicalModule::OperationalStatus <> PreviousInstance.CIM_LogicalModule::OperationalStatus.
QueryLanguage		Mandatory	This shall be 'DMTF:CQL' for CQL queries.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

15.8.7 CIM_IndicationFilter (WQL Blades LogicalModule OperationalStatus)

Deprecated. This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of a blade LogicalModule instance. This would represent the OperationalStatus changes of a blade LogicalModule. This is a special case of the CIM_IndicationFilter (pre-defined) class as defined in the Indication Profile.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 178 describes class CIM_IndicationFilter (WQL Blades LogicalModule OperationalStatus).

**Table 178 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(WQL Blades LogicalModule OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	See the SystemCreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
CreationClassName		Mandatory	See the CreationClassName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SystemName		Mandatory	See the SystemName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
Name		Mandatory	This shall be 'SNIA:Blades:LogicalModuleOperationalStatusWQL'.
SourceNamespace	N	Optional	Deprecated. See the SourceNamespace definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).
SourceNamespaces	N	Mandatory	Experimental. See the SourceNamespaces definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

**Table 178 - SMI Referenced Properties/Methods for CIM_IndicationFilter
(WQL Blades LogicalModule OperationalStatus)**

Properties	Flags	Requirement	Description and Notes
Query		Mandatory	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalModule AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus.
QueryLanguage		Mandatory	This shall be 'WQL' or 'SMI-S V1.0'.
ElementName	N	Optional	See the ElementName definition in ISO/IEC 24775-3, <i>Information Technology – Storage Management, Part 3 Common Profiles</i> 43.8.4 CIM_IndicationFilter (pre-defined).

15.8.8 CIM_LogicalModule

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 179 describes class CIM_LogicalModule.

Table 179 - SMI Referenced Properties/Methods for CIM_LogicalModule

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	Opaque.
ElementName		Mandatory	
OperationalStatus		Mandatory	
ModuleNumber		Mandatory	

15.8.9 CIM_MemberOfCollection (Blades FilterCollection to Switch FilterCollection)

Experimental. This associates the Blades predefined FilterCollection to the Switch predefined FilterCollection.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 180 describes class CIM_MemberOfCollection (Blades FilterCollection to Switch FilterCollection).

**Table 180 - SMI Referenced Properties/Methods for CIM_MemberOfCollection
(Blades FilterCollection to Switch FilterCollection)**

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the Switch predefined FilterCollection.
Member		Mandatory	Reference to the Blades predefined FilterCollection.

15.8.10 CIM_MemberOfCollection (Predefined Filter Collection to Blades Filters)

Experimental. This associates the Blades predefined FilterCollection to the predefined Filters supported by the blade.

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 181 describes class CIM_MemberOfCollection (Predefined Filter Collection to Blades Filters).

Table 181 - SMI Referenced Properties/Methods for CIM_MemberOfCollection (Predefined Filter Collection to Blades Filters)

Properties	Flags	Requirement	Description and Notes
Collection		Mandatory	Reference to the Blades predefined FilterCollection.
Member		Mandatory	Reference to the predefined IndicationFilters of the blade.

15.8.11 CIM_ModulePort

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 182 describes class CIM_ModulePort.

Table 182 - SMI Referenced Properties/Methods for CIM_ModulePort

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	A reference to a switch FCPort.
GroupComponent		Mandatory	A reference to a LogicalModule.

15.8.12 CIM_PhysicalPackage (Logical Module)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 183 describes class CIM_PhysicalPackage (Logical Module).

Table 183 - SMI Referenced Properties/Methods for CIM_PhysicalPackage (Logical Module)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	
Tag		Mandatory	
ElementName		Optional	
Name		Optional	
Manufacturer		Mandatory	

Table 183 - SMI Referenced Properties/Methods for CIM_PhysicalPackage (Logical Module)

Properties	Flags	Requirement	Description and Notes
Model		Mandatory	
SerialNumber		Optional	
Version		Optional	
PartNumber		Optional	

15.8.13 CIM_Product (Blade)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 184 describes class CIM_Product (Blade).

Table 184 - SMI Referenced Properties/Methods for CIM_Product (Blade)

Properties	Flags	Requirement	Description and Notes
Name		Mandatory	Commonly used Product name.
IdentifyingNumber		Mandatory	Product identification such as a serial number.
Vendor		Mandatory	The manufacturer or the OEM.
Version		Mandatory	Product version information.
ElementName		Mandatory	User-friendly name. Suggested use is Vendor, Version and product name.

15.8.14 CIM_ProductPhysicalComponent

Associates the Product to the PhysicalPackage. This is necessary to link the Product information to the Blade.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 185 describes class CIM_ProductPhysicalComponent.

Table 185 - SMI Referenced Properties/Methods for CIM_ProductPhysicalComponent

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	
PartComponent		Mandatory	

15.8.15 CIM_Realizes (Logical Module Package)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Optional.

Table 186 describes class CIM_Realizes (Logical Module Package).

Table 186 - SMI Referenced Properties/Methods for CIM_Realizes (Logical Module Package)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	The reference to the PhysicalPackage.
Dependent		Mandatory	The reference to the LogicalModule representing the Blade.

15.8.16 CIM_SystemDevice (Logical Module)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 187 describes class CIM_SystemDevice (Logical Module).

Table 187 - SMI Referenced Properties/Methods for CIM_SystemDevice (Logical Module)

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	A reference to a Switch ComputerSystem.
PartComponent		Mandatory	A reference to a LogicalModule.

STABLE

EXPERIMENTAL

16 Switch Partitioning Subprofile

16.1 Description

The Switch Partitioning Subprofile is used when a “switch” actually is implementing multiple instances of a profile. The instances of the profile can be the same profiles, for example the Switch Profile, or different profiles, for example the Switch Profile and the Extender Profile. For the context of further discussion, the “switch” representing the entire set of systems will be called the Partitioning System and the systems that it is “hosting” are the Partitioned System. For virtual fabrics, FC-SW-5 calls the partitioning system the “Core Switch” and the partitioned system the “Virtual Switch”. In other literature, a Partitioning System may be referred to as a Physical System or Physical Switch and a Partitioned Switch may be called a Logical Switch. In Figure 26, an example is shown with a Partitioning System hosting a Partitioned System running the Extender Profile and the Switch Profile. The nomenclature in this profile uses blue to identify the partitioning entity.

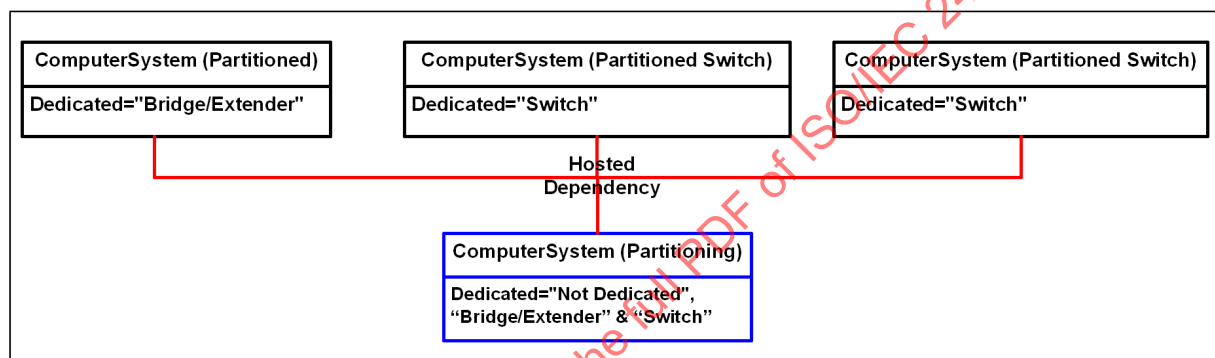


Figure 26 - Switch ComputerSystem and Partitioning System

The relationship between the Partitioning System and its partitions is indicated with the HostedDependency association. The Partitioning System can be determined by locating the ComputerSystem that is not a dependent in a HostedDependency relationship to another ComputerSystem. The Partitioning System also has the Dedicated property set to “Not Dedicated” plus the Dedicated values of the partitioned systems it supports.

In Figure 27, the Partitioning Ports have been added. For this Profile, the NetworkPort class or a subclass is required to identify the partitioning entity.

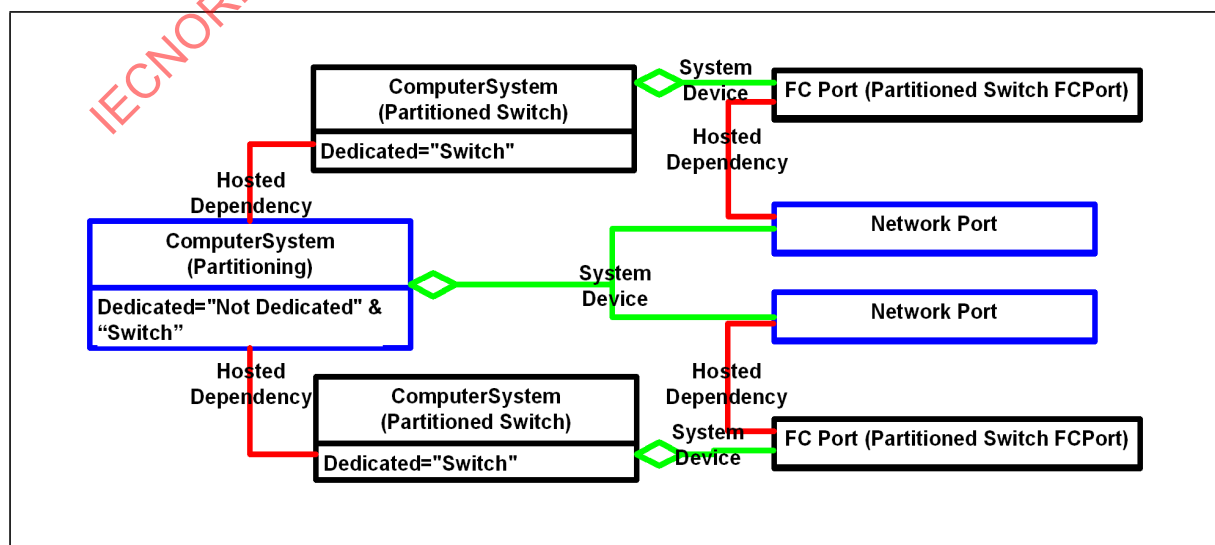


Figure 27 - Switch and Partitioning System and Partitioning Ports

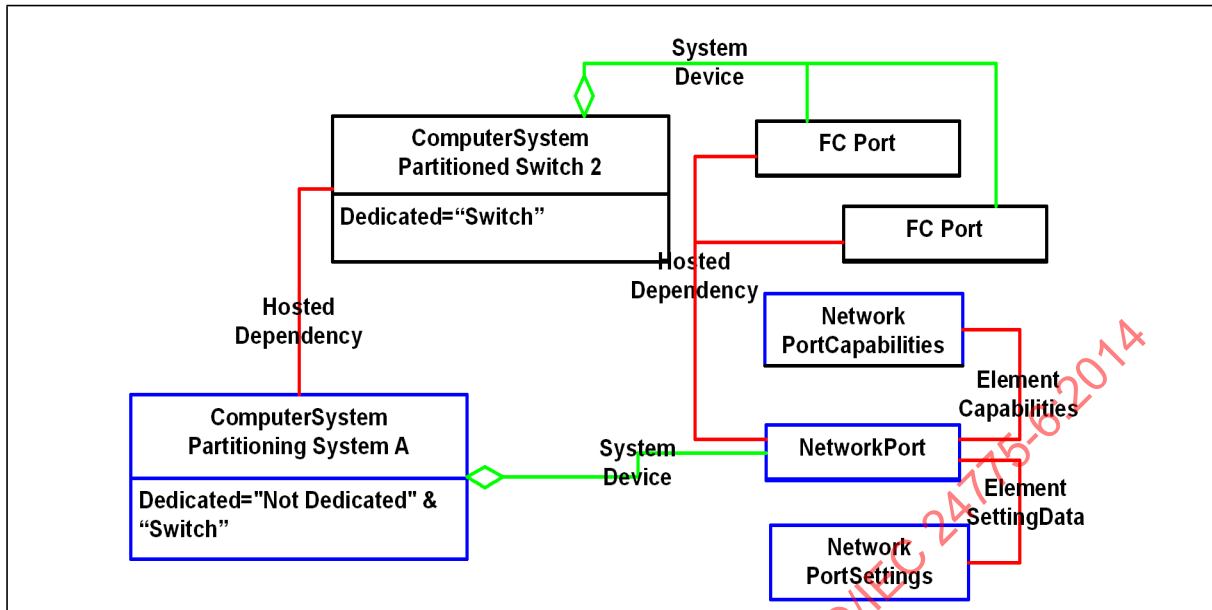


Figure 28 - Underlying System Port Settings and Capabilities

In Figure 28, there is the class `NetworkPortCapabilities` with the property `NetworkIDsConfigurable` which indicates whether this port can be assigned to a particular fibre channel fabric. If this property is true, then the property, `NetworkIDsFormat` should be `VF_ID` and the array property `NetworkIDs` can be set to define which fibre channel virtual fabrics this port can belong to. When the port connects to one of the fibre channel fabrics in the setting, an `FCPort` instance shall be instantiated to represent the Fibre Channel Port that is active in the particular fabric with the association `HostedDependency` showing which `NetworkPort` the `FCPort` was partitioned from. If the setting contained more than one `NetworkID`, for every active connection to a fabric, an instance of `FCPort` shall be instantiated.

The `FCSwitches` partitioned from the partitioning `ComputerSystem` are implicitly created when a `FCPort` is connected to a virtual fabric. Currently there is no mechanism to explicitly create a partitioned switch.

The `Physical Package` is supported as part of this subprofile and shall be associated to the partitioning system. The partitioned systems may also be associated to the `Physical Package` if the partitioned system is defined in a separate profile/subprofile. For example, the partitioned system in the `Switch Profile` may associate to the same instances of `PhysicalPackage` and `Product` that the partitioning system is associated to.

16.2 Health and Fault Management Consideration

None.

16.3 Cascading Considerations

None.

16.4 Supported Profiles, Subprofiles, and Packages

Table 188 describes the supported profiles for `FabricSwitchPartitioning`.

Table 188 - Supported Profiles for `FabricSwitchPartitioning`

Profile Name	Organization	Version	Requirement	Description
Physical Package	SNIA	1.5.0	Mandatory	

16.5 Methods of the Profile

None.

16.6 Client Considerations and Recipes

None.

16.7 Registered Name and Version

FabricSwitchPartitioning version 1.5.0 (Component Profile).

16.8 CIM Elements**16.8.1 General**

Table 189 describes the CIM elements for FabricSwitchPartitioning.

Table 189 - CIM Elements for FabricSwitchPartitioning

Element Name	Requirement	Description
16.8.2 CIM_ComputerSystem (Partitioned)	Mandatory	The partitioned ComputerSystem.
16.8.3 CIM_ComputerSystem (Partitioning)	Mandatory	The partitioning ComputerSystem.
16.8.4 CIM_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities)	Mandatory	Association between NetworkPort and NetworkPortCapabilities.
16.8.5 CIM_ElementConformsToProfile (Partitioning ComputerSystem to Switch Partitioning RegisteredProfile)	Mandatory	Ties the Partitioning ComputerSystem to the registered profile for Switch Partitioning.
16.8.6 CIM_ElementSettingData (Association between NetworkPort and NetworkPortSettings)	Mandatory	Association between NetworkPort and NetworkPortSettings.
16.8.7 CIM_FCPort (Partitioned)	Mandatory	Fibre Channel Port on the partitioned system.
16.8.8 CIM_HostedDependency (NetworkPort to FCPort)	Mandatory	Association between NetworkPort to FCPort.
16.8.9 CIM_HostedDependency (Partitioning CS to Partitioned CS)	Mandatory	Association between the Partitioning ComputerSystem and Partitioned ComputerSystem.
16.8.10 CIM_NetworkPort (Partitioning)	Mandatory	The partitioning port.
16.8.11 CIM_NetworkPortCapabilities	Mandatory	The NetworkPort Capabilities.
16.8.12 CIM_NetworkPortSettings	Mandatory	Defines the Virtual Fabrics the switch port can connect to.
16.8.13 CIM_SystemDevice (FCPort to Partitioned ComputerSystem)	Mandatory	Associates the partitioned classes (FCPort to the ComputerSystem).
16.8.14 CIM_SystemDevice (NetworkPort to ComputerSystem)	Mandatory	Associates the partitioning classes (NetworkPort to the ComputerSystem).

16.8.2 CIM_ComputerSystem (Partitioned)

The ComputerSystem representing the Interconnect Element (e.g. a switch) or Platform (e.g. Host and Array).

Created By: External

Modified By: Static

Deleted By: External

Requirement: Mandatory.

Table 190 describes class CIM_ComputerSystem (Partitioned).

Table 190 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioned)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name	C	Mandatory	The Partitioned System Name. The name shall be correlatable to the system represented (e.g., a Switch).
Dedicated		Mandatory	
OtherIdentifyingInfo		Conditional	Conditional requirement: Required if the Virtual Fabrics profile is implemented. One of the entries shall contain the Virtual Fabric ID for the virtual fabric (if virtual fabrics is implemented). Other entries in OtherIdentifyingInfo define the partitioned system (e.g., Switch OtherIdentifyingInfo if the Partitioned system is a Switch).
IdentifyingDescriptions		Conditional	Conditional requirement: Required if the Virtual Fabrics profile is implemented. One of the entries shall contain 'SNIA:VF_ID' for the virtual fabric (if virtual fabrics is implemented). Other entries in IdentifyingDescriptions define the partitioned system (e.g., Switch IdentifyDescriptions if the Partitioned system is a Switch).

16.8.3 CIM_ComputerSystem (Partitioning)

The ComputerSystem representing the Interconnect Element (e.g. a switch) or Platform (e.g. Host and Array).

Created By: External

Modified By: Static

Deleted By: External

Requirement: Mandatory.

Table 191 describes class CIM_ComputerSystem (Partitioning).

Table 191 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Partitioning)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	Name of Class.
Name		Mandatory	The Partitioning System Name.
Dedicated		Mandatory	

16.8.4 CIM_ElementCapabilities

(Association between NetworkPort and NetworkPortCapabilities)

Association between NetworkPort and NetworkPortCapabilities.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 192 describes class CIM_ElementCapabilities (Association between NetworkPort and NetworkPortCapabilities).

**Table 192 - SMI Referenced Properties/Methods for CIM_ElementCapabilities
(Association between NetworkPort and NetworkPortCapabilities)**

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	The reference to the NetworkPort.
Capabilities		Mandatory	The reference to the NetworkPortCapabilities.

16.8.5 CIM_ElementConformsToProfile (Partitioning ComputerSystem to Switch Partitioning RegisteredProfile)

The CIM_ElementConformsToProfile ties Partitioning ComputerSystem to the registered profile for Switch Partitioning.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 193 describes class CIM_ElementConformsToProfile (Partitioning ComputerSystem to Switch Partitioning RegisteredProfile).

**Table 193 - SMI Referenced Properties/Methods for CIM_ElementConformsToProfile
(Partitioning ComputerSystem to Switch Partitioning RegisteredProfile)**

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	A Partitioning ComputerSystem that represents the base system supporting partitioned systems.
ConformantStandard		Mandatory	RegisteredProfile instance describing the Switch Partitioning profile.

16.8.6 CIM_ElementSettingData (Association between NetworkPort and NetworkPortSettings)

Association between NetworkPort and NetworkPortSettings.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 194 describes class CIM_ElementSettingData (Association between NetworkPort and NetworkPortSettings).

**Table 194 - SMI Referenced Properties/Methods for CIM_ElementSettingData
(Association between NetworkPort and NetworkPortSettings)**

Properties	Flags	Requirement	Description and Notes
SettingData		Mandatory	The reference to the NetworkPortSettings.
ManagedElement		Mandatory	The reference to the NetworkPort.

16.8.7 CIM_FCPort (Partitioned)

Fibre Channel Port on the partitioned system (e.g., a switch partition).

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 195 describes class CIM_FCPort (Partitioned).

Table 195 - SMI Referenced Properties/Methods for CIM_FCPort (Partitioned)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.
ElementName		Optional	Port Symbolic Name if available. Otherwise NULL. If the underlying implementation includes characters that are illegal in CIM strings, then truncate before the first of those characters.
PermanentAddress	CD	Mandatory	Fibre Channel Port WWN. Expressed as 16 un-separated upper case hex digits (see Table 4 for more information about formats).
NetworkAddresses	C	Mandatory	Fibre Channel ID (FCID). Expressed as 8 un-separated upper case hex digits (see Table 4 for more information about formats).
OperationalStatus		Mandatory	The operational status of the FC Port.
PortType		Mandatory	The specific port type currently enabled (from FC-GS-6 Port.Type).
LinkTechnology		Mandatory	Shall be 4 ('FC').
SupportedFC4Types		Optional	An array of integers indicating the Fibre Channel FC-4 protocols supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '27' (FC-SB-2 Channel) for Host Ports or '28' (FC-SB-2 Control Unit) for targets.] EXPERIMENTAL .
SupportedCOS		Optional	An array of integers indicating the Fibre Channel Classes of Service that are supported. EXPERIMENTAL [When supporting FC-SB, this property shall be either '2' or '3' for FC-SB-2 Channel or FC-SB-2 Control Unit ports, respectively.] EXPERIMENTAL .

16.8.8 CIM_HostedDependency (NetworkPort to FCPort)

The association between the partitioning NetworkPort and the partitioned FCPort. The Antecedent references the partitioning port and the Dependent references the partitioned port. The association can be used to determine whether the system is in the underlying SAN topology or the Fabric topology.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 196 describes class CIM_HostedDependency (NetworkPort to FCPort).

Table 196 - SMI Referenced Properties/Methods for CIM_HostedDependency (NetworkPort to FCPort)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	NetworkPort representing the Partitioning System.
Dependent		Mandatory	FCPort representing the Partitioned System.

16.8.9 CIM_HostedDependency (Partitioning CS to Partitioned CS)

The association between the Partitioning ComputerSystem and the Partitioned ComputerSystem. The Antecedent references the partitioning system and the Dependent references the partitioned system. The association can be used to determine whether the system is in the underlying SAN topology or the Fabric topology.

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 197 describes class CIM_HostedDependency (Partitioning CS to Partitioned CS).

Table 197 - SMI Referenced Properties/Methods for CIM_HostedDependency (Partitioning CS to Partitioned CS)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	ComputerSystem representing the Partitioning System.
Dependent		Mandatory	ComputerSystem representing the Partitioned System.

16.8.10 CIM_NetworkPort (Partitioning)

The partitioning port.

Created By: External

Modified By: Static

Deleted By: External

Requirement: Mandatory.

Table 198 describes class CIM_NetworkPort (Partitioning).

Table 198 - SMI Referenced Properties/Methods for CIM_NetworkPort (Partitioning)

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	The scoping System's CreationClassName.
SystemName		Mandatory	The scoping System's Name.
CreationClassName		Mandatory	Name of Class.
DeviceID		Mandatory	Opaque.

16.8.11 CIM_NetworkPortCapabilities

The NetworkPort Capabilities.

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory.

Table 199 describes class CIM_NetworkPortCapabilities.

Table 199 - SMI Referenced Properties/Methods for CIM_NetworkPortCapabilities

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	Shall be set to "NetworkPortCapabilities".
NetworkIDsConfigurable		Mandatory	If the switch supports configuring virtual fabrics, this property shall be TRUE. If the switch only supports discovery, this property shall be FALSE.
NetworkIDsFormat		Mandatory	For configuring virtual fabrics, this property shall be for VF_ID.

16.8.12 CIM_NetworkPortSettings

Defines the Virtual Fabrics the switch port can connect to. The property NetworkIDs is an array which should contain the Virtual Fabric IDs (VF_ID) that the NetworkPort will partition FCPorts for.

Created By: Static
 Modified By: Static
 Deleted By: Static
 Requirement: Mandatory.

Table 200 describes class CIM_NetworkPortSettings.

Table 200 - SMI Referenced Properties/Methods for CIM_NetworkPortSettings

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	Shall be set to "NetworkPortSettings".
NetworkIDs		Mandatory	Contains the Virtual Fabric IDs that the NetworkPort will host FCPorts for.

16.8.13 CIM_SystemDevice (FCPort to Partitioned ComputerSystem)

Associates the FCPort to the partitioned ComputerSystem.

Created By: External
 Modified By: Static
 Deleted By: External
 Requirement: Mandatory.

Table 201 describes class CIM_SystemDevice (FCPort to Partitioned ComputerSystem).

**Table 201 - SMI Referenced Properties/Methods for CIM_SystemDevice
(FCPort to Partitioned ComputerSystem)**

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	The reference to the partitioned ComputerSystem.
PartComponent		Mandatory	The reference to the partitioned FCPort.

16.8.14 CIM_SystemDevice (NetworkPort to ComputerSystem)

Associates the NetworkPort to the ComputerSystem.

Created By: External

Modified By: Static

Deleted By: External

Requirement: Mandatory.

Table 202 describes class CIM_SystemDevice (NetworkPort to ComputerSystem).

**Table 202 - SMI Referenced Properties/Methods for CIM_SystemDevice
(NetworkPort to ComputerSystem)**

Properties	Flags	Requirement	Description and Notes
GroupComponent		Mandatory	The reference to the ComputerSystem.
PartComponent		Mandatory	The reference to the NetworkPort.

EXPERIMENTAL

EXPERIMENTAL

17 Extender Profile

17.1 Description

17.1.1 General

A FC Extender is a logical entity representing an inter-switch link consisting of two FC Extender Node devices and the Network pipes that connect them.

A FC Extender is used to connect two Fabrics across a LAN, MAN, WAN, or other network communications media.

A FC Extender Node is a physical device that converts Fibre Channel protocol for transmission over different network communication technologies.

The domain of the Extender Group is defined by Network, which is a subclass of AdminDomain.

17.1.2 FC Extender Node Topology Classes

The ComputerSystem class is the core of the model. It is identified as an Extender node by the dedicated attribute being set to ExtenderNode.

The TCPSettings and IPSettings classes represent the global configuration of the FC Extender transport layer.

The Port group of classes contains the following classes: FCPort, and EthernetPort. The FCPort class represents the connection of a FC Extender to a SAN. This class connects to other FCPort classes to represent Fibre channel connections. This class could be replaced with other port types to represent SANs based on other interconnect technology. The EthernetPort class represents an Ethernet link between FC Extender nodes. Figure 29 shows an FC Extender Node instance.

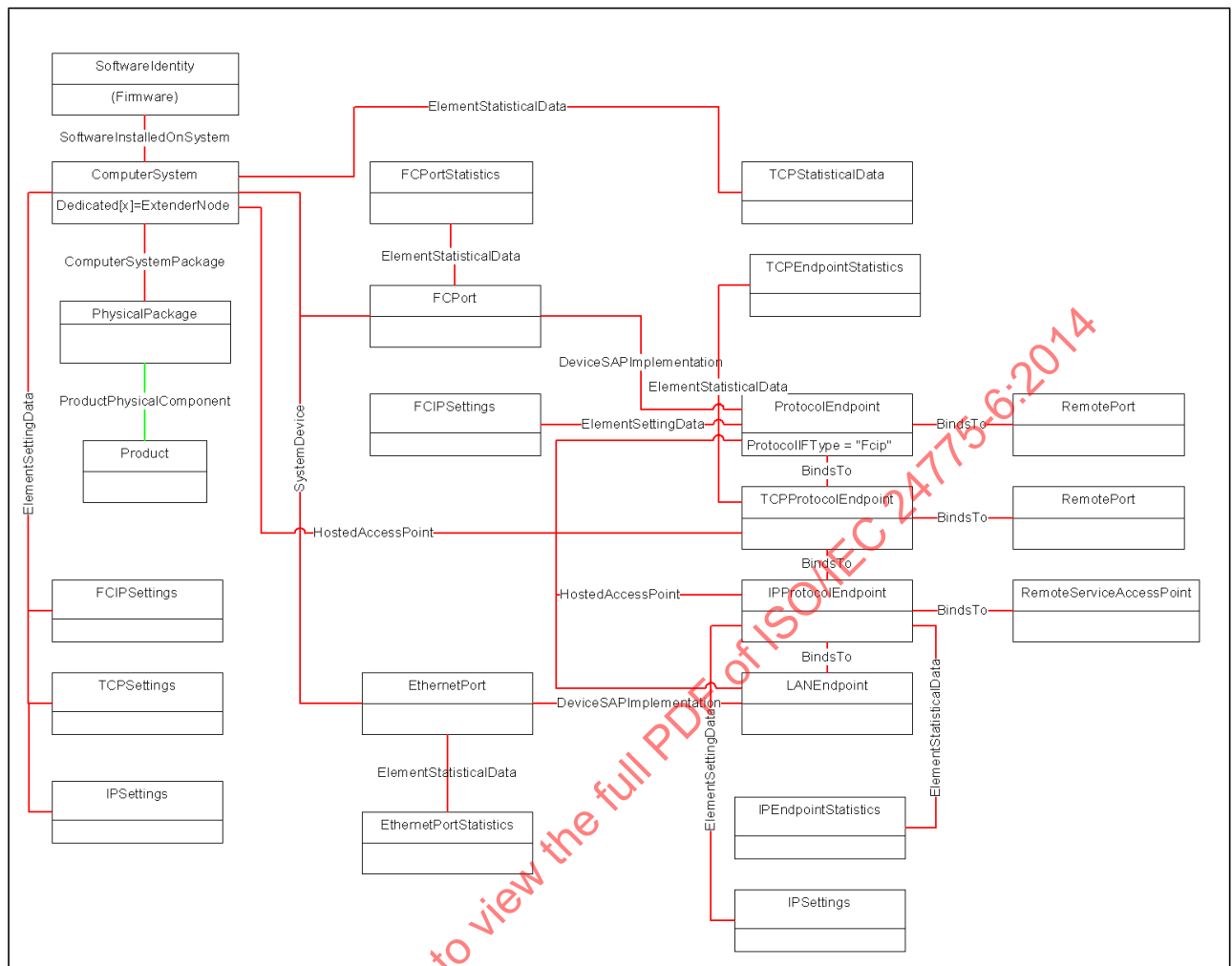


Figure 29 - FC Extender Node Instance

17.1.3 FC Extender Node Network Connectivity Classes

Each FC Extender node local `ProtocolEndpoint` (e.g., `FCProtocolEndpoints`, `TCPProtocolEndpoints`) has a `BindsTo` dependency on a `RemotePort` that describes access or addressing information to a remote `ProtocolEndpoint` for a specific connection.

The Extender node represents `ProtocolEndpoints` dependencies (e.g., `FC ProtocolEndpoint` on `TCPProtocolEndpoints`, `TCPProtocolEndpoint` on `IPProtocolEndpoint`, `IPProtocolEndpoint` on `EthernetProtocolEndpoint`) with a `BindsTo` association.

17.1.4 FC Extender Group Network Connectivity Classes

A FC Extender connection is represented by a `NetworkPipe` class associated with `FCProtocolEndpoints`. An `FCExtender Network` class groups multiple `NetworkPipes`.

The `NetworkPipe` between `FCProtocolEndpoints` is composed of lower-level TCP network pipes. Figure 30 shows a FC Extender Group instance.

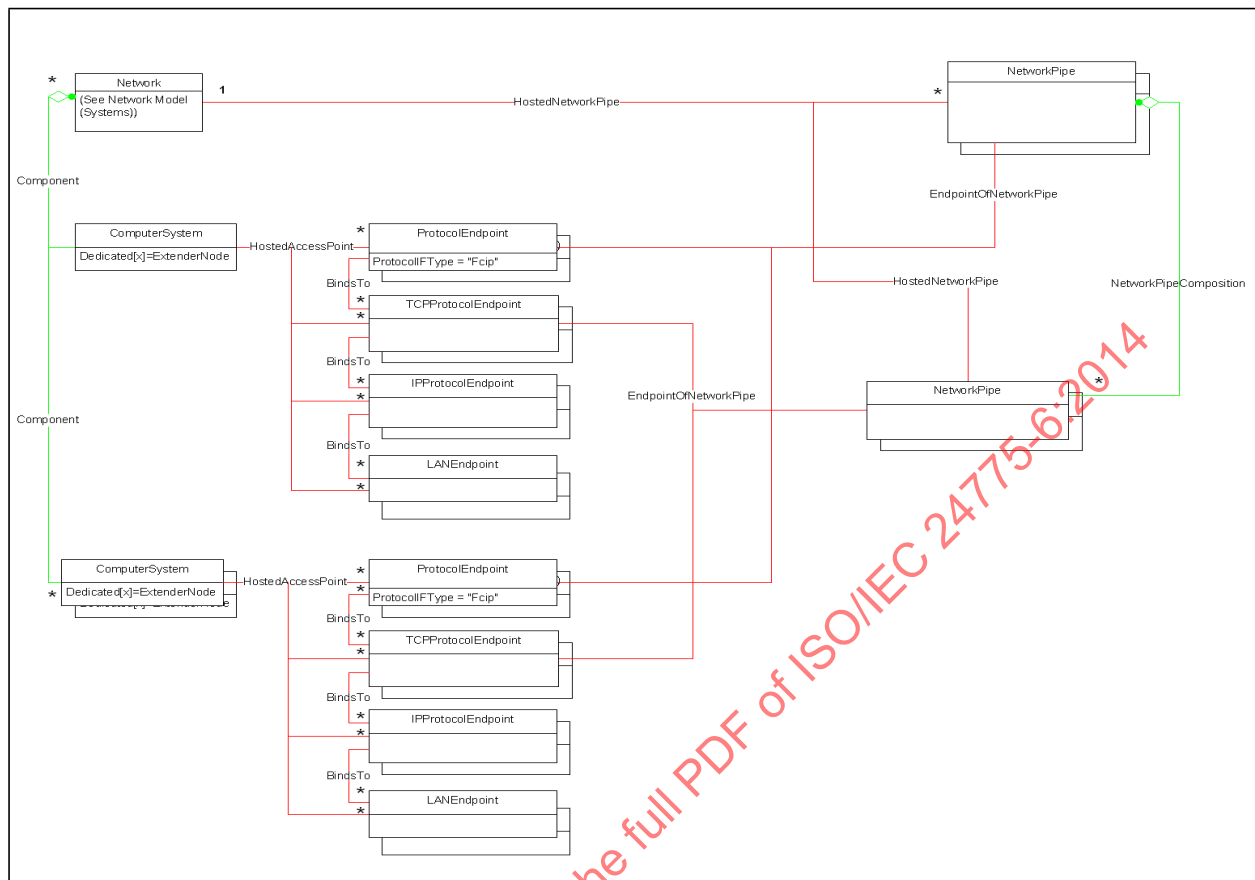


Figure 30 - FC Extender Group Instance

EXPERIMENTAL

17.1.5 Extender Predefined Indications

If the optional Experimental Indication profile is supported by an implementation of the Extender profile, it shall implement the `SNIA_IndicationConfigurationService` and its associated `SNIA_IndicationConfigurationCapabilities`. If the implementation supports predefined `IndicationFilters` or predefined `IndicationFilterCollections` this shall be indicated in the `SupportedFeatures` property of the `SNIA_IndicationConfigurationCapabilities`. If a value "3" is present, it means the implementation supports predefined `IndicationFilters`. If a value of "5" is present, it means the implementation supports predefined `IndicationFilterCollections`.

Figure 31 illustrates classes that shall be populated by the Extender profile if both "3" and "5" are present in the `SupportedFeatures` property.

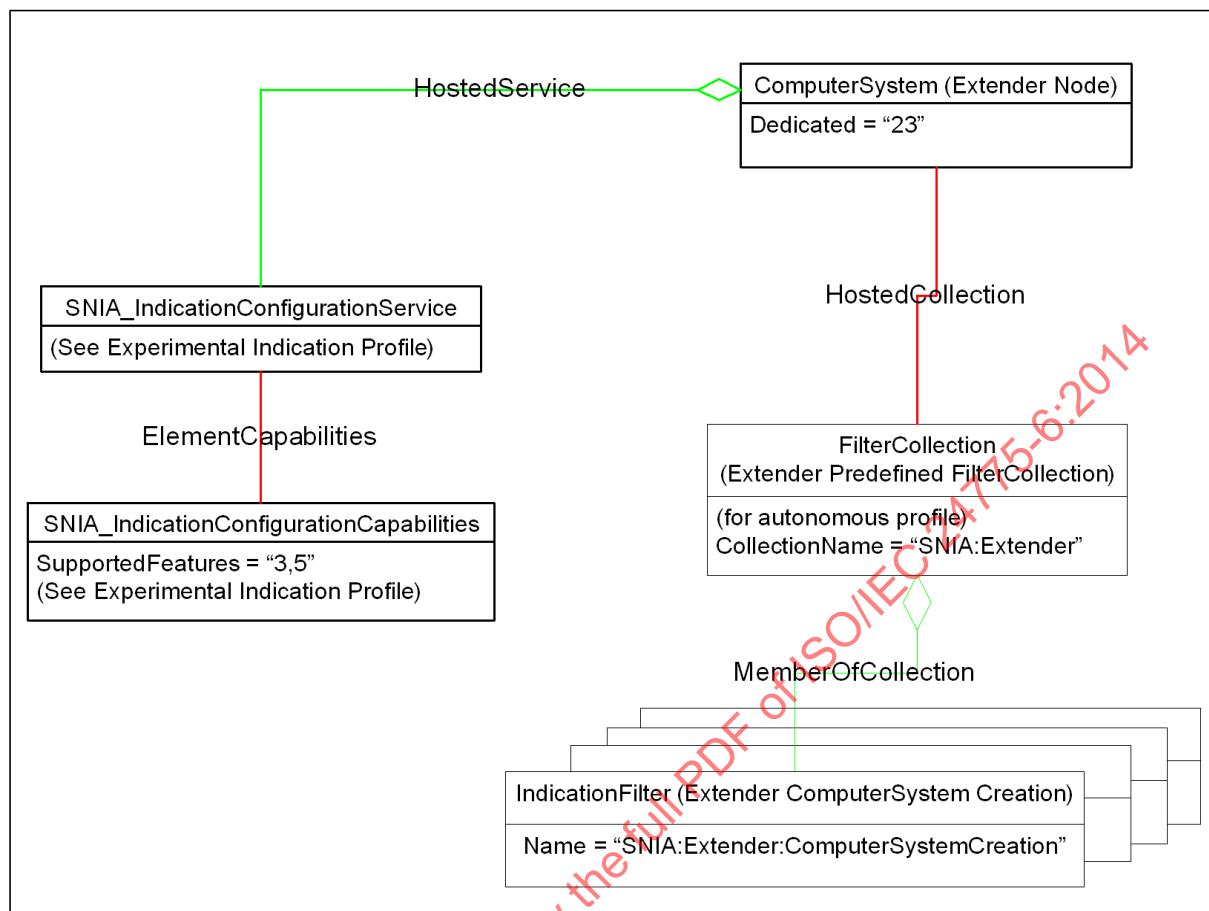


Figure 31 - Predefined Extender IndicationFilters and Collection

The SNIA_IndicationConfigurationService is hosted on the extender node ComputerSystem. The FilterCollection for the Extender is also hosted on the extender node ComputerSystem. The Extender FilterCollection has members which are all the predefined IndicationFilters supported by the implementation. This shall include all Mandatory IndicationFilters of the Extender Profile. But it should also include any IndicationFilter that has been predefined by the implementation. This may include conditional, optional or vendor specific IndicationFilters supported by the implementation.

The Extender FilterCollection shall have the CollectionName "SNIA:Extender". Each of the predefined filters shall have the Name property as defined for the IndicationFilter. In Figure 31 the name of the IndicationFilter (Extender ComputerSystem Creation) is "SNIA:Extender:ComputerSystemCreation". For vendor specific IndicationFilters (not defined in this International Standard), the Name of the filter would be of the form ORG_ID:Extender:UNIQUE_ID, where ORGID is the designation of the vendor that is providing the implementation.

EXPERIMENTAL

17.2 Health and Fault Management

None.

17.3 Cascading Considerations

None.

17.4 Supported Subprofiles and Packages

Table 203 describes the supported profiles for Extender.

Table 203 - Supported Profiles for Extender

Profile Name	Organization	Version	Requirement	Description
Physical Package	SNIA	1.5.0	Mandatory	
Software	SNIA	1.4.0	Mandatory	
Indication	SNIA	1.5.0	Mandatory	
Experimental Indication	SNIA	1.5.0	Optional	Experimental.

17.5 Methods of this Profile

None.

17.6 Client Considerations and Recipes

17.6.1 Extender Connectivity Settings

```
// Description
// Collecting settings
// of Extender node connectivity elements participating in the Extender
// of interest.
// $extenderNodeFCIPSettings(fcip protocol endpoint settings)
// $extenderNodeTCPSettings (transport layer settings)
// $extenderIPSettings (ip protocol endpoint settings)
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The Extender fcip ProtocolEndpoint has been previously
// identified and defined in the $fcipProtocolEndpoint-> variable
// 1. Get ComputerSystem associated with $fcipProtocolEndpoint
```

```
$extenderNodes[] = Associators(
    $fcipProtocolEndpoint->,
    CIM_HostedAccessPoint,
    CIM_ComputerSystem,
    Antecedent,
    Dependent,
    false,
    false,
    [Dedicated])
if (contains(23, $extenderNodes[0].Dedicated))
{
    #extenderNodeAccessA = true
    $extenderNode-> = $extenderNodes[0].getObjectPath()
}
if(#extenderNodeAccessA)
{
    //2. Get fcip protocol endpoint

    // find FCIP Settings
    $fcipSettings[] = Associators(
        $fcipProtocolEndpoint,
```

```

        CIM_ElementSettingData,
        CIM_FCIPSettings,
        ManagedElement,
        SettingData,
        false,
        false,
        null)
    if ($fcipSettings[].length != 0)
        $extenderNodeFCIPSettings = $fcipSettings[0]

```

```
//3. Get transport layer settings
```

```

    $tcpSettings[] = Associators(
        $extenderNode->,
        CIM_ElementSettingData,
        CIM_TCPSettings,
        ManagedElement,
        SettingData,
        false,
        false,
        null)
    if ($tcpSettings[].length != 0)
        $extenderNodeTCPSettings = $tcpSettings[0]

```

```
//4. Find TCPProtocolEndpoint bound to the extender fcip ProtocolEndpoint
```

```

    $tcpProtocolEndpoint->[] = AssociatorNames(
        $fcipProtocolEndpoint->,
        CIM_BindTo,
        CIM_TCPProtocolEndpoint,
        Dependent,
        Antecedent,
        false,
        false,
        null))

```

```
//5. Find IPProtocolEndpoint bound to the extender tcp ProtocolEndpoint
```

```

    $ipProtocolEndpoints->[] = AssociatorNames(
        $tcpProtocolEndpoint->,
        CIM_BindTo,
        CIM_IPProtocolEndpoint,
        Dependent,
        Antecedent,
        false,
        false,
        null))

```

```
$ipProtocolEndpoint-> = $ipProtocolEndpoints->[0]

//6. Find IPProtocolEndpoint settings

$ipSettings[] = Associators(
    $ipProtocolEndpoint->,
    CIM_ElementSettingData,
    CIM_IPSettings,
    ManagedElement,
    SettingData,
    false,
    false,
    null)
if ($ipSettings[].length != 0)
    $extenderNodeIPSettings = $ipSettings[0]

}
```

17.6.2 Extender Connective Statistics

```
// Description
// Collecting statistical data
// of Extender node connectivity elements participating in the Extender
// of interest.
// $extenderNodeTCPStatisticalData (transport layer stats)
// $extenderIPEndpointStatistics (IP protocol endpoint stats)
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The Extender fcip ProtocolEndpoint has been previously
// identified and defined in the $fcipProtocolEndpoint-> variable
// 1. Get ComputerSystem associated with $fcipProtocolEndpoint

$extenderNodes[] = Associators(
    $fcipProtocolEndpoint->,
    CIM_HostedAccessPoint,
    CIM_ComputerSystem,
    Antecedent,
    Dependent,
    false,
    false,
    [Dedicated])
if (contains(23, $extenderNodes[0].Dedicated))
{
    #extenderNodeAccess = true
    $extenderNode-> = $extenderNodes[0].getObjectPath()
}
if(#extenderNodeAccess)
{
    //2. Get transport layer statistics
```

```

$tcpStatistics[] = Associators(
    $extenderNode->,
    CIM_ElementStatisticalData,
    CIM_TCPStatisticalData,
    ManagedElement,
    Stats,
    false,
    false,
    null))
$extenderNodeTCPStatisticalData = $tcpStatistics[0]

```

//3. Find TCPProtocolEndpoint bound to the extender fcip ProtocolEndpoint

```

$tcpProtocolEndpoint->[] = AssociatorNames(
    $fcipProtocolEndpoint->,
    CIM_BindTo,
    CIM_TCPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))

```

//4. Find IPProtocolEndpoint bound to the extender tcp ProtocolEndpoint

```

$ipProtocolEndpoint->[] = AssociatorNames(
    $tcpProtocolEndpoint->,
    CIM_BindTo,
    CIM_IPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))

```

//5. Find IPProtocolEndpoint statistics

```

$ipStatistics[] = Associators(
    $ipProtocolEndpoint->,
    CIM_ElementStatisticalData,
    CIM_IPEndpointStatistics,
    ManagedElement,
    Stats,
    false,
    false,
    null)

```

```
$extenderIPEndpointStatistics = $ipStatistics[0]

}
```

17.6.3 Extender Port Group Information

```
// Description
// Collecting configuration and statistical data
// of Extender node ports participating in the Extender
// of interest.
// $extenderNodeFCPort (connected to a switch)
// $extenderNodeFCPortStatistics
// $extenderNodeEthernetPort (connected to a peer Extender node)
// $extenderNodeEthernetStatistics
// PREEXISTING CONDITIONS AND ASSUMPTIONS
// The Extender fcip ProtocolEndpoint has been previously
// identified and defined in the $fcipProtocolEndpoint-> variable

// 1. Get ComputerSystem associated with $fcipProtocolEndpoint

$extenderNodes[] = Associators(
    $fcipProtocolEndpoint->,
    CIM_HostedAccessPoint,
    CIM_ComputerSystem,
    Antecedent,
    Dependent,
    false,
    false,
    [Dedicated])
if (contains(23, $extenderNodes[0].Dedicated))
{
    #extenderNodeAccess = true
}
if (#extenderNodeAccess)
{
    // 2. Get FC port

    $fcPorts[] = Associators(
        $fcipProtocolEndpoints->,
        CIM_DeviceSAPImplementation,
        CIM_FCPort,
        Dependent,
        Antecedent,
        false,
        false,
        null)
    $extenderNodeFCPort = $fcPorts[0]

    // 2. Get FC port statistics
```

```

$fcPortStatistics->[] = Associators(
    $extenderNodeFCPort.getObjectPath(),
    CIM_ElementStatisticalData,
    CIM_FCPortStatistics,
    ManagedElement,
    Stats,
    false,
    false,
    null))
$extenderNodeFCPortStatistics = $fcPortsStatistics[0]

```

```

//3. Find TCPProtocolEndpoint bound to the extender FCIP
    ProtocolEndpoint

```

```

$tcpProtocolEndpoints->[] = AssociatorNames(
    $fcipProtocolEndpoint->,
    CIM_BindTo,
    CIM_TCPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))
// at least one should exist
$tcpProtocolEndpoint->=$tcpProtocolEndpoints->[0]

```

```

//4. Find IPProtocolEndpoint bound to the extender TCP ProtocolEndpoint

```

```

$ipProtocolEndpoint->[] = AssociatorNames(
    $tcpProtocolEndpoint->,
    CIM_BindTo,
    CIM_IPProtocolEndpoint,
    Dependent,
    Antecedent,
    false,
    false,
    null))
$ipProtocolEndpoint->=$ipProtocolEndpoints->[0]

```

```

//5. Get Ethernet port

```

```

$ethernetPorts[] = Associators(
    $ipProtocolEndpoints->,
    CIM_DeviceSAPImplementation,
    CIM_EthernetPort,
    Dependent,
    Antecedent,
    false,
    false,
    null)

```

```
$extenderNodeEthernetPort = $ethernetPorts[0]

//6. Get Ethernet port statistics

$ethernetPortStatistics->[] = Associators(
    $extenderNodeEthernetPort.getObjectPath(),
    CIM_ElementStatisticalData,
    CIM_EthernetStatistics,
    ManagedElement,
    Stats,
    false,
    false,
    null))
$extenderNodeEthernetPortStatistics = $ethernetPortStatistics[0]

}
```

17.6.4 Extender Topology Mapping

```
// This recipe describes how to build a topology graph of a fabric.
//
// 1. Identifies all the Switches and adds their objects paths and the
// object paths of the FC Ports belonging to these Switches to the $nodes
// array
//
// 2. Creates a suitable Association instance (e.g., a SystemDevice
// Association instance between a Switch and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array
//
// 3. Creates a map of all connected FC Ports (i.e., belonging to
// Switches
// that are ISL'd together and to Host HBAs and Storage System Front End
// Controllers)
//
// In this map, the FC Ports (i.e., the ones that are connected) are
// cross-connected.
//
// e.g., For a pair of FC Ports, one belonging to a Switch and the other
// belonging to a Host (HBA), the map indexed by the Switch Port WWN
// returns
// the Host (HBA) FC Port object path and the map indexed by the Host
// (HBA)
// FC Port WWN returns the Switch FC Port object path.
//
// Similar relationship exists between the pairs of FC Ports where one
// belongs to a Switch and the other belonging belongs to a Storage
// System
// Front End Controller and for FC Ports each of which belongs to a
// Switch.
//
// 4. Identifies all the Hosts and adds their objects paths to the $nodes
```

```

// array. Note that the object paths of the FC Ports (HBA Ports)
// belonging
// to these Hosts are already added to the $nodes array in step-3.
//
// 5. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Host and a FC Port), setting its
// GroupComponent and PartComponent. Adds the object path of the
// Association
// to the $links array.
//
// 6. Identifies all the Storage Systems and adds their objects paths to
// the
// $nodes array.
// Note that the object paths of the FC Ports (i.e., Front End Controller
// FC Ports) belonging to these Storage Systems are already added to the
// $nodes array in step-3.
//
// 7. Creates a suitable Association instance (e.g. a SystemDevice
// Association instance between a Storage System and a FC Port), setting
// its GroupComponent and PartComponent. Adds the object path of the
// Association to the $links array.

// DESCRIPTION
// Create a map of how elements in a SAN are connected together via
// Fibre-ChannelFC ports.
//
// The map is built in array $attachedFcPorts->[], where the index is a
// WWN of any device port on the SAN, and the value at that index is
// the object path of the connected Switch or HBA or Storage System FC
// port.
//
// First find all the switches in a SAN. Get all the FC Ports for each
// switch and get the Attached FC Ports for each Switch FC Port. Save
// these
// device FC ports in the map described above.

// PREEXISTING CONDITIONS AND ASSUMPTIONS
// 1. All agents/namespaces supporting Fabric Profile previously
// identified
// using SLP. Do this for each CIMOM supporting Fabric Profile

switches[] = enumerateInstances("CIM_ComputerSystem", true, false, true,
// true, null)

for #i in $switches[]
{
    if (!contains(5, $switches[#i].Dedicated))
        continue

    // only process switches, not other computer systems

    // Add the switch to the $nodes array

```

```
$nodes.addIfNotAlreadyAdded ($switches[#i].getObjectPath());

// Get all the SystemDevice associations between this switch and its
// FC Ports

$sysDevAssoc[] = ReferenceNames($switches[#i],
                                "CIM_FCPort",
                                "GroupComponent");

// Add these associations to the $links array

for #a in $sysDevAssoc->[]
$links.addIfNotAlreadyAdded ($sysDevAssoc->[#a];

$fcPorts->[] = AssociatorNames(
    $switches[#i].getObjectPath(),
    "CIM_SystemDevice",
    "CIM_FCPort",
    "GroupComponent",
    "PartComponent")
for #j in $fcPorts->[]
{

    // Add the FC Port in $nodes array

    $nodes.addIfNotAlreadyAdded (fcPorts->[#j];

    $protocolEndpoints->[] = AssociatorNames(
        fcPorts->[#j],
        "CIM_DeviceSAPImplementation",
        "CIM_ProtocolEndpoint",
        "Antecedent",
        "Dependent");

    // NOTE - It is possible for this collection to be empty (i.e.,
    // ports
    // that are not connected). It is possible for this collection to
    // have more than one element (loops attached to a switch port is
    // the
    // most common example).

    if ($protocolEndpoints->[].length == 0)
        continue

    // Add the Protocol End Point to the nodes array.
    // Currently this recipe is designed to only save one
    // ProtocolEndpoint.

    $nodes.addIfNotAlreadyAdded (protocolEndpoints[0]);

    // Add the associations between the fcPort and the Protocol end
    // point
```



```
// Add the attached FC Port to the $nodes array
if $attachedFcPort != null
    $nodes.addIfNotAlreadyAdded ($attachedFcPort);
}
}
}
```

17.7 Registered Name and Version

Extender version 1.5.0 (Autonomous Profile).

17.8 CIM Elements

17.8.1 General

Table 204 describes the CIM elements for Extender.

Table 204 - CIM Elements for Extender

Element Name	Requirement	Description
17.8.2 CIM_BindsTo (IPPE to PE)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.3 CIM_BindsTo (IPPE to RemoteSAP)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.4 CIM_BindsTo (IPPE to RemoteSAP)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.5 CIM_BindsTo (PE to RemotePort)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.6 CIM_BindsTo (TCPPE to IPPE)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.7 CIM_BindsTo (TCPPE to PE)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.8 CIM_BindsTo (TCPPE to RemotePort)	Mandatory	Associates Extender Node ProtocolEndpoints from different layers in the protocol stack.
17.8.9 CIM_Component	Mandatory	Aggregates Extender Nodes in the Network that represents the group of Extenders.
17.8.10 CIM_ComputerSystem (Extender Node)	Mandatory	Represents the Extender Node Associated to RegisteredProfile.
17.8.11 CIM_DeviceSAPImplementation	Mandatory	
17.8.12 CIM_ElementSettingData (IPPE to IPSettings)	Mandatory	Associates SettingData to Extender Node or ProtocolEndpoints.
17.8.13 CIM_ElementSettingData (PE to FCIPSettings)	Mandatory	Associates SettingData to Extender Node or ProtocolEndpoints.
17.8.14 CIM_ElementSettingData (System to FCIPSettings)	Mandatory	Associates SettingData to Extender Node or ProtocolEndpoints.
17.8.15 CIM_ElementSettingData (System to IPSettings)	Mandatory	Associates SettingData to Extender Node or ProtocolEndpoints.
17.8.16 CIM_ElementSettingData (System to TCPSettings)	Mandatory	Associates SettingData to Extender Node or ProtocolEndpoints.

Table 204 - CIM Elements for Extender

Element Name	Requirement	Description
17.8.17 CIM_ElementSettingData (TCPPE to TCPSettings)	Mandatory	Associates SettingData to Extender Node or ProtocolEndpoints.
17.8.18 CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics)	Mandatory	Associates StatisticalData to Extender Node or ProtocolEndpoints.
17.8.19 CIM_ElementStatisticalData (FCPort to FCPortStatistics)	Mandatory	
17.8.20 CIM_ElementStatisticalData (IPPE to IPEndpointStatistics)	Mandatory	
17.8.21 CIM_ElementStatisticalData (System to TCPStatisticalData)	Mandatory	
17.8.22 CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics)	Mandatory	
17.8.23 CIM_EndpointOfNetworkPipe (PE to NetworkPipe)	Mandatory	
17.8.24 CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe)	Mandatory	
17.8.25 CIM_EthernetPort	Mandatory	
17.8.26 CIM_EthernetPortStatistics	Mandatory	
17.8.27 CIM_FCIPSettings	Mandatory	Defines FCIP settings for a group of ProtocolEndpoints (ProtocolIFType - "Fcip") which belongs to the ComputerSystem (Extender Node).
17.8.28 CIM_FCPort	Mandatory	
17.8.29 CIM_FCPortStatistics	Mandatory	
17.8.30 CIM_FilterCollection (Extender Predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This is a collection of predefined IndicationFilters to which a client may subscribe.
17.8.31 CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)	Mandatory	Associates the ProtocolEndpoint to the ComputerSystem or Network.
17.8.32 CIM_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint)	Mandatory	Associates the ProtocolEndpoint to the ComputerSystem or Network.
17.8.33 CIM_HostedAccessPoint (ComputerSystem to ProtocolEndpoint)	Mandatory	Associates the ProtocolEndpoint to the ComputerSystem or Network.
17.8.34 CIM_HostedCollection (Extender to predefined FilterCollection)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).
17.8.35 CIM_HostedNetworkPipe	Mandatory	Associates NetworkPipe to the Network.
17.8.36 CIM_IPEndpointStatistics	Mandatory	
17.8.37 CIM_IPProtocolEndpoint	Mandatory	
17.8.38 CIM_IPSettings	Mandatory	Defines IP settings for a group of IPProtocolEndpoints which belongs to the ComputerSystem.

Table 204 - CIM Elements for Extender

Element Name	Requirement	Description
17.8.39 CIM_IndicationFilter (Extender ComputerSystem Creation)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the addition of a new Extender Node computer system instance.
17.8.40 CIM_IndicationFilter (Extender ComputerSystem Deletion)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the removal of an Extender Node computer system instance.
17.8.41 CIM_IndicationFilter (Extender ComputerSystem OperationalStatus)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the 'pre-defined' CIM_IndicationFilter instance for the modification of OperationalStatus of an Extender Node computer system instance.
17.8.42 CIM_IndicationFilter (WQL Extender ComputerSystem OperationalStatus)	Conditional	Deprecated. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='3' (Predefined Filters). This is the WQL 'pre-defined' CIM_IndicationFilter instance for OperationalStatus changes of an Extender Node ComputerSystem instance.
17.8.43 CIM_LANEndpoint	Mandatory	The ProtocolEndpoint for the Ethernet port.
17.8.44 CIM_MemberOfCollection (Predefined Filter Collection to Extender Filters)	Conditional	Experimental. Conditional requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections). This associates the Extender predefined FilterCollection to the predefined Filters supported by the Extender.
17.8.45 CIM_Network	Mandatory	Network represents a network connectivity domain. It groups NetworkPipes.
17.8.46 CIM_NetworkPipe	Mandatory	NetworkPipe represents state, configuration of a connection between endpoints in the context of a Network.
17.8.47 CIM_NetworkPipeComposition	Mandatory	
17.8.48 CIM_ProtocolEndpoint	Mandatory	The ProtocolEndpoint representing the FCIP layer. ProtocolEndpoint shall be implemented when an ActiveConnection or NetworkPipe exists. It may be implemented if no ActiveConnection or NetworkPipe exists.
17.8.49 CIM_RemotePort	Optional	This class shall be implemented if the instrumentation is only able to support the FC Extender Node.

Table 204 - CIM Elements for Extender

Element Name	Requirement	Description
17.8.50 CIM_RemoteServiceAccessPoint	Optional	This class shall be implemented if the instrumentation is only able to support the FC Extender Node.
17.8.51 CIM_SystemDevice (System to EthernetPort)	Mandatory	Associated FCPort and EthernetPort to the ComputerSystem.
17.8.52 CIM_SystemDevice (System to FCPort)	Mandatory	Associated FCPort and EthernetPort to the ComputerSystem.
17.8.53 CIM_TCPEndpointStatistics	Mandatory	Opaque.
17.8.54 CIM_TCPProtocolEndpoint	Mandatory	
17.8.55 CIM_TCPSettings	Mandatory	Defines TCP settings for a group of TCPProtocolEndpoints which belongs to the ComputerSystem.
17.8.56 CIM_TCPStatisticalData	Mandatory	
SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem	Mandatory	Creation of a ComputerSystem instance. See 17.8.39 CIM_IndicationFilter (Extender ComputerSystem Creation).
SELECT * FROM CIM_InstDeletion WHERE SourceInstance CIM_ComputerSystem	Mandatory	Deletion of a computer system instance. See 17.8.40 CIM_IndicationFilter (Extender ComputerSystem Deletion).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.OperationalStatus <> PreviousInstance.OperationalStatus	Mandatory	Deprecated. Deprecated WQL - Change of OperationalStatus for a Computer System. See 17.8.42 CIM_IndicationFilter (WQL Extender ComputerSystem OperationalStatus).
SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus	Mandatory	CQL - Change of OperationalStatus for a Computer System. See 17.8.41 CIM_IndicationFilter (Extender ComputerSystem OperationalStatus).

17.8.2 CIM_BindsTo (IPPE to PE)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 205 describes class CIM_BindsTo (IPPE to PE).

Table 205 - SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to PE)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	
Dependent		Mandatory	

17.8.3 CIM_BindsTo (IPPE to RemoteSAP)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 206 describes class CIM_BindsTo (IPPE to RemoteSAP).

Table 206 - SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to RemoteSAP)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	
Antecedent		Mandatory	

17.8.4 CIM_BindsTo (IPPE to RemoteSAP)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 207 describes class CIM_BindsTo (IPPE to RemoteSAP).

Table 207 - SMI Referenced Properties/Methods for CIM_BindsTo (IPPE to RemoteSAP)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	
Dependent		Mandatory	

17.8.5 CIM_BindsTo (PE to RemotePort)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 208 describes class CIM_BindsTo (PE to RemotePort).

Table 208 - SMI Referenced Properties/Methods for CIM_BindsTo (PE to RemotePort)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	
Dependent		Mandatory	

17.8.6 CIM_BindsTo (TCPPE to IPPE)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 209 describes class CIM_BindsTo (TCPPE to IPPE).

Table 209 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to IPPE)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	
Dependent		Mandatory	

17.8.7 CIM_BindsTo (TCPPE to PE)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 210 describes class CIM_BindsTo (TCPPE to PE).

Table 210 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to PE)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	
Dependent		Mandatory	

17.8.8 CIM_BindsTo (TCPPE to RemotePort)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 211 describes class CIM_BindsTo (TCPPE to RemotePort).

Table 211 - SMI Referenced Properties/Methods for CIM_BindsTo (TCPPE to RemotePort)

Properties	Flags	Requirement	Description and Notes
Antecedent		Mandatory	
Dependent		Mandatory	

17.8.9 CIM_Component

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 212 describes class CIM_Component.

Table 212 - SMI Referenced Properties/Methods for CIM_Component

Properties	Flags	Requirement	Description and Notes
PartComponent		Mandatory	
GroupComponent		Mandatory	

17.8.10 CIM_ComputerSystem (Extender Node)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Shall be associated to RegisteredProfile using ElementConformsToProfile association. The RegisteredProfile instance shall have RegisteredName set to 'Extender', RegisteredOrganization set to 'SNIA', and RegisteredVersion set to '1.5.0'.

Table 213 describes class CIM_ComputerSystem (Extender Node).

Table 213 - SMI Referenced Properties/Methods for CIM_ComputerSystem (Extender Node)

Properties	Flags	Requirement	Description and Notes
CreationClassName		Mandatory	
Name		Mandatory	IP Address.
ElementName		Optional	User friendly name.
NameFormat		Mandatory	IP Address.
OtherIdentifyingInfo		Optional	DNS name.
OperationalStatus		Mandatory	Status of Computer System.
IdentifyingDescriptions		Optional	Fully qualified domain name.
Dedicated		Mandatory	ExtenderNode.

17.8.11 CIM_DeviceSAPImplementation

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 214 describes class CIM_DeviceSAPImplementation.

Table 214 - SMI Referenced Properties/Methods for CIM_DeviceSAPImplementation

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	
Antecedent		Mandatory	
Dependent		Mandatory	
Antecedent		Mandatory	

17.8.12 CIM_ElementSettingData (IPPE to IPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 215 describes class CIM_ElementSettingData (IPPE to IPSettings).

Table 215 - SMI Referenced Properties/Methods for CIM_ElementSettingData (IPPE to IPSettings)

Properties	Flags	Requirement	Description and Notes
IsDefault		Mandatory	
IsCurrent		Mandatory	
ManagedElement		Mandatory	
SettingData		Mandatory	

17.8.13 CIM_ElementSettingData (PE to FCIPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 216 describes class CIM_ElementSettingData (PE to FCIPSettings).

Table 216 - SMI Referenced Properties/Methods for CIM_ElementSettingData (PE to FCIPSettings)

Properties	Flags	Requirement	Description and Notes
IsDefault		Mandatory	
IsCurrent		Mandatory	
SettingData		Mandatory	
ManagedElement		Mandatory	

17.8.14 CIM_ElementSettingData (System to FCIPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 217 describes class CIM_ElementSettingData (System to FCIPSettings).

Table 217 - SMI Referenced Properties/Methods for CIM_ElementSettingData (System to FCIPSettings)

Properties	Flags	Requirement	Description and Notes
IsDefault		Mandatory	
IsCurrent		Mandatory	
SettingData		Mandatory	
ManagedElement		Mandatory	

17.8.15 CIM_ElementSettingData (System to IPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 218 describes class CIM_ElementSettingData (System to IPSettings).

Table 218 - SMI Referenced Properties/Methods for CIM_ElementSettingData (System to IPSettings)

Properties	Flags	Requirement	Description and Notes
IsDefault		Mandatory	
IsCurrent		Mandatory	
SettingData		Mandatory	
ManagedElement		Mandatory	

17.8.16 CIM_ElementSettingData (System to TCPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 219 describes class CIM_ElementSettingData (System to TCPSettings).

Table 219 - SMI Referenced Properties/Methods for CIM_ElementSettingData (System to TCPSettings)

Properties	Flags	Requirement	Description and Notes
IsDefault		Mandatory	
IsCurrent		Mandatory	
SettingData		Mandatory	
ManagedElement		Mandatory	

17.8.17 CIM_ElementSettingData (TCPPE to TCPSettings)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 220 describes class CIM_ElementSettingData (TCPPE to TCPSettings).

Table 220 - SMI Referenced Properties/Methods for CIM_ElementSettingData (TCPPE to TCPSettings)

Properties	Flags	Requirement	Description and Notes
IsDefault		Mandatory	
IsCurrent		Mandatory	
SettingData		Mandatory	
ManagedElement		Mandatory	

17.8.18 CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 221 describes class CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics).

Table 221 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (EthernetPort to EthernetPortStatistics)

Properties	Flags	Requirement	Description and Notes
Stats		Mandatory	
ManagedElement		Mandatory	

17.8.19 CIM_ElementStatisticalData (FCPort to FCPortStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 222 describes class CIM_ElementStatisticalData (FCPort to FCPortStatistics).

Table 222 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (FCPort to FCPortStatistics)

Properties	Flags	Requirement	Description and Notes
Stats		Mandatory	
ManagedElement		Mandatory	

17.8.20 CIM_ElementStatisticalData (IPPE to IPEndpointStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 223 describes class CIM_ElementStatisticalData (IPPE to IPEndpointStatistics).

Table 223 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (IPPE to IPEndpointStatistics)

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	
Stats		Mandatory	

17.8.21 CIM_ElementStatisticalData (System to TCPStatisticalData)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 224 describes class CIM_ElementStatisticalData (System to TCPStatisticalData).

Table 224 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (System to TCPStatisticalData)

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	
Stats		Mandatory	

17.8.22 CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 225 describes class CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics).

Table 225 - SMI Referenced Properties/Methods for CIM_ElementStatisticalData (TCPPE to TCPEndpointStatistics)

Properties	Flags	Requirement	Description and Notes
ManagedElement		Mandatory	
Stats		Mandatory	

17.8.23 CIM_EndpointOfNetworkPipe (PE to NetworkPipe)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 226 describes class CIM_EndpointOfNetworkPipe (PE to NetworkPipe).

Table 226 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (PE to NetworkPipe)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	
Antecedent		Mandatory	

17.8.24 CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 227 describes class CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe).

Table 227 - SMI Referenced Properties/Methods for CIM_EndpointOfNetworkPipe (TCPPE to NetworkPipe)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	
Antecedent		Mandatory	

17.8.25 CIM_EthernetPort

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 228 describes class CIM_EthernetPort.

Table 228 - SMI Referenced Properties/Methods for CIM_EthernetPort

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
ElementName		Optional	User-friendly name.
OperationalStatus		Mandatory	
Speed		Mandatory	
MaxSpeed		Mandatory	
PortType		Mandatory	Supported port mode 10BaseT, 10-100BaseT, 100BaseT, 1000BaseT, etc.

Table 228 - SMI Referenced Properties/Methods for CIM_EthernetPort

Properties	Flags	Requirement	Description and Notes
PortNumber		Mandatory	System level port or bus identification number.
NetworkAddresses		Mandatory	MAC addresses.
LinkTechnology		Mandatory	Ethernet.

17.8.26 CIM_EthernetPortStatistics

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 229 describes class CIM_EthernetPortStatistics.

Table 229 - SMI Referenced Properties/Methods for CIM_EthernetPortStatistics

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Mandatory	
StatisticTime		Optional	
BytesTransmitted		Mandatory	
BytesReceived		Mandatory	
PacketsTransmitted		Mandatory	
PacketsReceived		Mandatory	
SymbolErrors		Mandatory	
AlignmentErrors		Optional	
FCSErrors		Optional	
SingleCollisionFrames		Optional	
MultipleCollisionFrames		Optional	
DeferredTransmissions		Optional	
LateCollisions		Optional	
ExcessiveCollisions		Optional	
InternalMACTransmitErrors		Optional	
InternalMACReceiveErrors		Optional	
CarrierSenseErrors		Mandatory	
FrameTooLongs		Optional	
ResetSelectedStats()		Optional	

17.8.27 CIM_FCIPSettings

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 230 describes class CIM_FCIPSettings.

Table 230 - SMI Referenced Properties/Methods for CIM_FCIPSettings

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Optional	User-friendly name. In addition, it can be used as a index property for a search or query.
ConnectionUsageFlags		Mandatory	
SpecialFrameTimeout		Mandatory	
KeepAliveTimeout		Mandatory	

17.8.28 CIM_FCPort

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 231 describes class CIM_FCPort.

Table 231 - SMI Referenced Properties/Methods for CIM_FCPort

Properties	Flags	Requirement	Description and Notes
SystemCreationClassName		Mandatory	
SystemName		Mandatory	
CreationClassName		Mandatory	
DeviceID		Mandatory	
ElementName		Optional	User-friendly name.
OperationalStatus		Mandatory	
Speed		Mandatory	Speed of zero represents a link not established 1 GB/s is 1 062 500 000 bit/s 2 GB/s is 2 125 000 000 bit/s 4 GB/s is 4 250 000 000 bit/s) 10 GB/s single channel variants are 10 518 750 000 bit/s 10 GB/s four channel variants are 12 750 000 000 bit/s This is the raw bit rate.
MaxSpeed		Mandatory	The maximum speed of the Port in bits per second using the same algorithm as Speed.

Table 231 - SMI Referenced Properties/Methods for CIM_FCPort

Properties	Flags	Requirement	Description and Notes
PortType		Mandatory	FC-GS-6 Port.Type The specific mode currently enabled for the Port. The values: "N" = Node Port, "NL" = Node Port supporting FC arbitrated loop, "E" = Expansion Port connecting fabric elements (for example, FC switches), "F" = Fabric (element) Port, "FL" = Fabric (element) Port supporting FC arbitrated loop, and "B" = Bridge Port. PortTypes are defined in the FC-GS-6 standards. When set to 1 ("Other"), the related property OtherPortType contains a string description of the port's type. PortType is defined to force consistent naming of the 'type' property in subclasses and to guarantee unique enum values for all instances of NetworkPort. A range of values, DMTF_Reserved, has been defined that allows subclasses to override and define their specific port types. Vendor Reserved = 16000..65535 can be used if the PortType is not one already defined in the above enumerations and a vendor subclass is defined specifying the appropriate 'value' and 'valuemap'.
PortNumber		Mandatory	System level port or bus identification number.
PermanentAddress		Mandatory	For FibreChannel, it is the Fibre Channel Port WWN.
LinkTechnology		Mandatory	"FC".
SupportedCOS		Mandatory	
ActiveCOS		Optional	
SupportedMaximumTransmissionUnit		Mandatory	
ActiveMaximumTransmissionUnit		Optional	

17.8.29 CIM_FCPortStatistics

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 232 describes class CIM_FCPortStatistics.

Table 232 - SMI Referenced Properties/Methods for CIM_FCPortStatistics

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Opaque.
ElementName		Optional	
StatisticTime		Mandatory	
BytesTransmitted		Mandatory	
BytesReceived		Mandatory	
PacketsTransmitted		Mandatory	
PacketsReceived		Mandatory	
CRRErrors		Mandatory	
LinkFailures		Mandatory	

Table 232 - SMI Referenced Properties/Methods for CIM_FCPortStatistics

Properties	Flags	Requirement	Description and Notes
PrimitiveSeqProtocolErrCount		Mandatory	
LossOfSignalCounter		Mandatory	
InvalidTransmissionWords		Mandatory	
SampleInterval		Mandatory	
LIPCount		Mandatory	
NOSCount		Mandatory	
ErrorFrames		Mandatory	
DumpedFrames		Mandatory	
LossOfSyncCounter		Mandatory	
FramesTooShort		Mandatory	
FramesTooLong		Mandatory	
AddressErrors		Mandatory	
BufferCreditNotProvided		Mandatory	
DelimiterErrors		Mandatory	
EncodingDisparityErrors		Mandatory	
LinkResetsReceived		Mandatory	
LinkResetsTransmitted		Mandatory	
MulticastFramesReceived		Mandatory	
MulticastFramesTransmitted		Mandatory	
RXBroadcastFrames		Mandatory	
TXBroadcastFrames		Mandatory	
FBSYFrames		Mandatory	
PBSYFrames		Mandatory	
FRJTFrames		Mandatory	
PRJTFrames		Mandatory	
RXClass1Frames		Mandatory	
TXClass1Frames		Mandatory	
RXClass2Frames		Mandatory	
TXClass2Frames		Mandatory	
Class2FBSY		Mandatory	
Class2PBSY		Mandatory	
Class2FRJT		Mandatory	
Class2PRJT		Mandatory	
RXClass3Frames		Mandatory	

Table 232 - SMI Referenced Properties/Methods for CIM_FCPortStatistics

Properties	Flags	Requirement	Description and Notes
TXClass3Frames		Mandatory	
Class3FramesDiscarded		Mandatory	
ResetSelectedStats()		Optional	

17.8.30 CIM_FilterCollection (Extender Predefined FilterCollection)

Experimental. This is a collection of predefined IndicationFilters to which a client may subscribe. An Extender implementation shall indicate support for predefined FilterCollections by the SNIA_IndicationConfigurationCapabilities.FeaturesSupported = '5' (Predefined Filter Collections).

Requirement: Required if the Experimental Indication Profile is supported and the SNIA_IndicationConfigurationCapabilities.SupportedFeatures='5' (Predefined Filter Collections).

Table 233 describes class CIM_FilterCollection (Extender Predefined FilterCollection).

Table 233 - SMI Referenced Properties/Methods for CIM_FilterCollection (Extender Predefined FilterCollection)

Properties	Flags	Requirement	Description and Notes
InstanceID		Mandatory	Shall specify the unique identifier for an instance of this class within the Implementation namespace.
CollectionName		Mandatory	The value of CollectionName shall be 'SNIA:Extender'.

17.8.31 CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.

Table 234 describes class CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint).

Table 234 - SMI Referenced Properties/Methods for CIM_HostedAccessPoint (ComputerSystem to IPProtocolEndpoint)

Properties	Flags	Requirement	Description and Notes
Dependent		Mandatory	
Antecedent		Mandatory	

17.8.32 CIM_HostedAccessPoint (ComputerSystem to TCPProtocolEndpoint)

Created By: Static

Modified By: Static

Deleted By: Static

Requirement: Mandatory.